View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Missouri University of Science and Technology (Missouri S&T): Scholars' Mine

Scholars' Mine

Doctoral Dissertations

Student Theses and Dissertations

Spring 2018

Incremental proper orthogonal decomposition for PDE simulation data: Algorithms and analysis

Hiba Fareed

Follow this and additional works at: https://scholarsmine.mst.edu/doctoral_dissertations

 Part of the Mathematics Commons

Department: Mathematics and Statistics

Recommended Citation

Fareed, Hiba, "Incremental proper orthogonal decomposition for PDE simulation data: Algorithms and analysis" (2018). *Doctoral Dissertations*. 2671.
https://scholarsmine.mst.edu/doctoral_dissertations/2671

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

INCREMENTAL PROPER ORTHOGONAL DECOMPOSITION FOR PDE SIMULATION DATA: ALGORITHMS AND ANALYSIS

by

HIBA GHASSAN FAREED

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

MATHEMATICS

2018

Approved by

Dr. John Singler, Advisor
Dr. Xiaoming He
Dr. Yanzhi Zhang
Dr. Robert Paige
Dr. Jennifer Leopold

Copyright 2018

HIBA GHASSAN FAREED

All Rights Reserved

PUBLICATION DISSERTATION OPTION

This dissertation consists of the two articles which have been submitted, or will be submitted for publication as follows:

Paper I: Pages 6-48 have been published in Journal of Computers and Mathematics with Applications.

Paper II: Pages 49-82 have been submitted to Journal of Computational and Mathematics with Applications, and a preprint has been posted online at https://arxiv.org/abs/1803.06313.

ABSTRACT

We propose an incremental algorithm to compute the proper orthogonal decomposition (POD) of simulation data for a partial differential equation. Specifically, we modify an incremental matrix SVD algorithm of Brand to accommodate data arising from Galerkin-type simulation methods for time dependent PDEs. We introduce an incremental SVD algorithm with respect to a weighted inner product to compute the proper orthogonal decomposition (POD). The algorithm is applicable to data generated by many numerical methods for PDEs, including finite element and discontinuous Galerkin methods. We also modify the algorithm to initialize and incrementally update both the SVD and an error bound during the time stepping in a PDE solver without storing the simulation data. We show the algorithm produces the exact SVD of an approximate data matrix, and the operator norm error between the approximate and exact data matrices is bounded above by the computed error bound. This error bound also allows us to bound the error in the incrementally computed singular values and singular vectors. We demonstrate the effectiveness of the algorithm using finite element computations for a 1D Burgers' equation, a 1D FitzHugh-Nagumo PDE system, and a 2D Navier-Stokes problem.

ACKNOWLEDGMENTS

I gratefully acknowledge HCED for giving me the scholarship and all financial support to accomplish my study in the United States of America. I would like to thank Department of Mathematics and Statistics in Missouri University of Science and Technology for providing me with the equipment and classes that helped me throughout my PhD studies.

A cordial thanks and special gratitude to my adviser Dr. John R. Singler for his tireless effort, guidance, encouragement, and kindness. His generosity in providing information to me helped me to bring this work towards a completion and this dissertation to the existence. I would like to express my great thanks to committee members namely- Dr.Yanzhi Zhang, Dr. Jennifer Leopold, Dr.Xiaoming He, and Dr. Robert Paige for serving on my committee.

All the gratitude and appreciation to my beloved husband Esam for his support and patience and to help me achieve success and getting my PhD degree. There are not enough words to praise all that he has been done for me. So, this thesis is heartily dedicated to him. Also, I would like to thank my two sweet little daughters Maryam and Reetal for their love and patience.

Last but not the least, I would like to thank my parents and my sister for their encouragements and prayers for me during my education and during my life in general. No acknowledgment would be complete without giving thanks to my mother-in-law and father-in-law for their prayers and support.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION

1. INTRODUCTION

Proper orthogonal decomposition (POD) is an optimal data approximation method: It produces a basis called POD modes that can be used to optimally reconstruct the data. POD was introduced to the turbulence community by Lumley in 1967 [1]. Before that time it was already invented in 1901 by Karl Pearson [2] under the name principal component analysis (PCA) as an analogue of the principal axis theorem in mechanics. POD has been developed by several people, therefore it is known under other names such as principal component analysis [3], and the Karhunen-Loève decomposition [4, 5]. Proper orthogonal decomposition (POD) has been widely used in many applications involving partial differential equations (PDEs) such as aeroelasticity [6], fluid dynamics [7, 8], feedback control [9], PDE constrained optimization and optimal control problems [10, 11], uncertainty quantification [12], and data assimilation [13, 14]. The most important point in the POD procedure is optimality: it provides the most efficient way of capturing the dominant components of a finite or infinite-dimensional data set by producing the minimum number of basis elements (called POD modes) to reconstruct the data. In other words, POD is an effective method for reducing high-dimensional data sets.

We formulate POD as a constrained minimization problem. Let X be a Hilbert space with the inner product $(\cdot,\cdot)_X$ and norm $\|\cdot\|_X$. Suppose snapshots $x_1, x_2, \ldots, x_n \in X$ are given, where $\dim X \geq n$. Define $Y = \operatorname{span}\{x_1, x_2, \ldots, x_n\} \subset X$, and let $\{y_j\}_{j=1}^n$ be any orthonormal basis for Y . Then we have

$$x_i = \sum_{j=1}^{n} (x_i, y_j)_X \, y_j, \quad i = 1, \ldots, n. \tag{1.1}$$

For any $k \in \mathbb{N}$ and $k < n$, our goal is to find the orthonormal basis $\{y_j\}_{j=1}^{k}$ (the POD modes) of rank k which is the solution of the following problem:

$$\min \sum_{i=1}^{n} \|x_i - \hat{x}_i\|_X^2, \quad \text{with } (y_i, y_j)_X = \delta_{ij}, \quad \text{for } 1 \le i, j \le k, \tag{1.2}$$

where the approximation of x_i is

$$\hat{x}_i = \sum_{j=1}^{k} (x_i, y_j)_X \, y_j. \tag{1.3}$$

Therefore, the POD modes are the orthonormal basis of Y minimizing the total mean square error between the snapshots x_i and their corresponding low order approximations $\hat{x}_i$.

There are many methods to compute the POD of a dataset; the most basic approaches rely on computing the eigenvalue decomposition or singular value decomposition (SVD) of a matrix constructed using the dataset. The method of snapshots introduced by Sirovich [15] is the most widely used approach to find the POD of large data sets (see, e.g., [16, 17, 18]). However, the SVD based technique can be used to obtain low-rank matrix approximations (to recall SVD see [19, 20, 21]). SVD has been applied in conjunction with various techniques such as proper orthogonal decomposition (POD), although this approach requires a large amount of data storage and it is computationally expensive. Researchers have proposed various methods to deal with the computational complexity and data storage issues for POD computations (see, e.g., [22, 23, 24, 25, 26]). One of the approaches is called *incremental SVD algorithms*; specifically, the SVD of a data set is initialized on a small amount of data and then updated as new data arrives. For more information see, e.g., [27, 28, 26, 29, 30, 31, 32]. However, the Hilbert space inner product has not been thoroughly explored in these works. It is important to consider the inner product in the POD calculations for applications (see, e.g., [33, 34, 35, 36]).

Incremental SVD methods have been used in combination with numerical methods for PDEs to compute POD modes during the time stepping without storing any of the simulation data. For examples of this approach, see, e.g., [37, 38, 39].

In this dissertation, we focus on computing the POD of a data set in a Hilbert space that is expressed using a collection of basis functions; therefore, the data can be generated by many numerical methods for PDEs, including finite element and discontinuous Galerkin methods. The main purpose of our work is to propose an incremental algorithm for the POD computations where we account for the Hilbert space inner product. We extend Brand's incremental matrix SVD algorithm [27] to accommodate the Hilbert space data expanded in basis elements. We show that the POD of the Hilbert space data expressed in term of a basis is equivalent to the POD of the vectors in $\mathbb{R}^m$ of the coefficient data with respect to a weighted inner product on $\mathbb{R}^m$. Therefore, we present an algorithm that incrementally computes the matrix SVD with respect to a weighted inner product. This algorithm involves truncations. We demonstrate the effectiveness of the truncation in the algorithm by computing the error bound incrementally without storing the whole data. We compute the error bound between the exact data matrix and the approximate truncated SVD of the data matrix. This error bound allows us to bound the error in the incrementally computed singular values and singular vectors.

1.1. BACKGROUND: SVD

Let X and Y be separable Hilbert spaces, and let $A : X \to Y$ be a compact linear operator with Hilbert adjoint operator $A^* : Y \to X$. The self-adjoint nonnegative compact operators AA^* and A^*A have nonnegative eigenvalues, and the square root of the eigenvalues are equal to the singular values μ_k of A. A singular value of A is equal to zero if an eigenvalue of AA^* or A^*A is equal to zero. Furthermore, the corresponding orthonormal basis of eigenvectors of AA^* is $\{\xi_k\} \subset Y$ and the corresponding orthonormal basis of eigenvectors of A^*A is $\{\eta_k\} \subset X$, thus the corresponding singular value expansions

can be defined as follows:

$$A\eta = \sum_{k\geq 1} \mu_k \, (\eta, \eta_k)_X \, \xi_k, \quad A^*\xi = \sum_{k\geq 1} \mu_k \, (\xi, \xi_k)_Y \, \eta_k,$$

for all $\eta \in X$ and $\xi \in Y$. This gives

$$A\eta_i = \mu_i \xi_i, \quad A^*\xi_i = \mu_i \eta_i, \quad \forall \mu_i > 0$$

We consider different Hilbert spaces such as $\mathbb{R}^k$ that denotes the space with the standard inner product $(x, y) = y^T x$, for all $x, y \in \mathbb{R}^k$ (i.e., the Euclidean inner product or dot product). Let $\mathbb{R}^m_M$ denote the Hilbert space $\mathbb{R}^m$ with M-weighted inner product $(x, y)_M = y^T M x$, for all $x, y \in \mathbb{R}^m$, where $M \in \mathbb{R}^{m\times m}$ is a symmetric positive definite square matrix.

1.2. OUTLINE

The organization of this dissertation is as follows: In paper I [40], we show that the POD of the Hilbert space data expressed in terms of a basis is equivalent to the POD of the vectors in $\mathbb{R}^m$ of the coefficient data with respect to a weighted inner product on $\mathbb{R}^m$. In this paper we consider two approaches to compute the incremental SVD with respect to the weighted inner product: (1) using a Cholesky factor of the weight matrix, and (2) without using a Cholesky factor. The first approach follows standard POD ideas for weighted inner products (see, e.g., [18]), but requires the computation of the Cholesky factor and also solving linear systems involving the Cholesky factor. In the second approach, we avoid these extra computations by directly extending Brand's incremental matrix SVD algorithm [27] to work with a weighted inner product on $\mathbb{R}^m$. In paper II [41], we show that we compute an error bound incrementally without storing the whole data set by extending the incremental SVD algorithm for a weighted inner product from paper I. This work also

displays an error analysis that discusses the effect of truncation at each step, and provides more insight into the accuracy of the algorithm with truncation and the choices of the tolerances. Finally, conclusions and future research ideas are presented in Section 2.

PAPER

I. INCREMENTAL PROPER ORTHOGONAL DECOMPOSITION FOR PDE SIMULATION DATA

Hiba Fareed[1], John R. Singler[1], Yangwen Zhang[1], and Jiguang Shen[2]

[1] Department of Mathematics and Statistics

Missouri University of Science and Technology

[2] School of Mathematics, University of Minnesota

ABSTRACT

We propose an incremental algorithm to compute the proper orthogonal decomposition (POD) of simulation data for a partial differential equation. Specifically, we modify an incremental matrix SVD algorithm of Brand to accommodate data arising from Galerkin-type simulation methods for time dependent PDEs. The algorithm is applicable to data generated by many numerical methods for PDEs, including finite element and discontinuous Galerkin methods. The algorithm initializes and efficiently updates the dominant POD eigenvalues and modes during the time stepping in a PDE solver without storing the simulation data. We prove that the algorithm without truncation updates the POD exactly. We demonstrate the effectiveness of the algorithm using finite element computations for a 1D Burgers' equation and a 2D Navier-Stokes problem.

Keywords: proper orthogonal decomposition, incremental algorithm, weighted norm, finite element method

1. INTRODUCTION

Proper orthogonal decomposition (POD) has been widely used in many applications involving partial differential equations such as aeroelasticity [1], fluid dynamics [2], feedback control [3], PDE constrained optimization and optimal control [4, 5], uncertainty quantification [6], and data assimilation [7, 8]. In the most basic form, POD is an optimal data approximation method: the POD of a dataset produces a basis (called POD modes) that can be used to optimally reconstruct the data. There are many methods to compute the POD of a dataset; the most basic approaches rely on computing the eigenvalue decomposition or singular value decomposition (SVD) of a matrix constructed using the dataset. The method of snapshots introduced by Sirovich [9] is commonly used to find POD eigenvalues and modes. For more information, see, e.g., [10, 11, 12].

For very large datasets, such as datasets arising from simulations of partial differential equations (PDEs), the basic approaches to POD computations become computationally expensive and require a large amount of data storage. Researchers have proposed various methods to deal with the computational complexity (see, e.g., [13, 14, 15, 16]), and both the computational complexity and data storage (see, e.g., [17] and the references therein). The latter class of methods are *incremental SVD algorithms*; specifically, the SVD is initialized on a small amount of data and then updated as new data becomes available. This type of incremental algorithm can be easily used in conjunction with a time stepping code for simulating a time dependent PDE; the POD eigenvalues and modes are updated during the time stepping without storing any of the simulation data. For examples of this approach, see, e.g., [18, 19, 20].

In this work, we focus on computing the POD of data arising from a Galerkin-type simulation of a PDE. Specifically, the data lies in a Hilbert space and are expressed using a collection of basis functions; therefore, the data can be generated using many numerical methods for PDEs, including finite element and discontinuous Galerkin methods. We extend Brand's incremental matrix SVD algorithm [21] to accommodate data of this type.

Specifically, we show that the POD of the Hilbert space data expressed in term of a basis is equivalent to the POD of the vectors in $\mathbb{R}^m$ of the coefficient data with respect to a weighted inner product on $\mathbb{R}^m$ (see Appendix 1). We consider two approaches to compute the incremental SVD with respect to the weighted inner product: (1) using a Cholesky factor of the weight matrix (Section 3), and (2) without using a Cholesky factor (Section 4). The first approach follows standard POD ideas for weighted inner products (see, e.g., [12]), but requires the computation of the Cholesky factor and also solving linear systems involving the Cholesky factor. In the second approach, we avoid these extra computations by directly extending Brand's incremental matrix SVD algorithm [21] to work with a weighted inner product on $\mathbb{R}^m$. We analyze an idealized version of the second approach that does not involve truncation and prove that it produces the exact SVD with respect to the weighted inner product.

We link the second approach together with (approximate) POD of time varying functions in Section 5, and then present numerical results in Section 4. For the numerical results, we consider computing the POD of finite element simulation data for a 1D Burgers' equation and a 2D Navier-Stokes equation. For the 1D problem and a coarse discretization of the 2D flow problem, we compare the result of the second incremental SVD approach to the true results and find excellent a greement. For a large-scale simulation of the 2D flow problem, we compute the POD without storing all the data by calculating the incremental SVD only. We present conclusions in Section 5.

2. BASIC DEFINITIONS AND CONCEPTS

We begin by introducing many important definitions and concepts needed throughout this work.

For notational convenience, we adopt Matlab notation herein. Given a vector $u \in \mathbb{R}^n$ and $r \leq n$, let $u(1:r)$ denote the vector of the first r components of u. Similarly, for a matrix $A \in \mathbb{R}^{m\times n}$, we let $A(p:q, r:s)$ denote the submatrix of A consisting of the entries of A from rows $p, \ldots, q$ and columns $r, \ldots, s$.

2.1. THE SVD WITH RESPECT TO A WEIGHTED INNER PRODUCT

Let $M \in \mathbb{R}^{m\times m}$ be a symmetric positive definite square matrix. Let $\mathbb{R}^m_M$ denote the Hilbert space $\mathbb{R}^m$ with M-weighted inner product, i.e.,

$$(x, y)_M = y^T M x \quad \text{for all } x, y \in \mathbb{R}^m.$$

Throughout this work, $\mathbb{R}^k$ without a subscript denotes the space with the standard inner product $(x, y) = y^T x$ (i.e., the Euclidean inner product or dot product).

We require two functional analytic concepts for a matrix $P \in \mathbb{R}^{m\times n}$ considered as a mapping $P : \mathbb{R}^n \to \mathbb{R}^m_M$: the Hilbert adjoint operator and the singular value decomposition.

First, the Hilbert adjoint operator of the matrix $P : \mathbb{R}^n \to \mathbb{R}^m_M$ is a matrix $P^* : \mathbb{R}^m_M \to \mathbb{R}^n$ satisfying

$$(Px, y)_M = (x, P^* y) \quad \text{for all } x \in \mathbb{R}^n \text{ and } y \in \mathbb{R}^m_M.$$

We can show $P^* = P^T M$ as follows:

$$(Px, y)_M = (x, P^* y) \quad \Rightarrow \quad y^T M\, Px = (P^* y)^T x \quad \Rightarrow \quad y^T M\, Px = y^T (P^*)^T x.$$

Since this holds for all x, y, we have $M\, P = (P^*)^T$ and therefore $P^* = P^T M$.

Second, since the matrix $P : \mathbb{R}^n \to \mathbb{R}^m_M$ is a compact linear operator, it has a singular value decomposition: the nonzero eigenvalues of $PP^* : \mathbb{R}^m_M \to \mathbb{R}^m_M$ and $P^*P : \mathbb{R}^n \to \mathbb{R}^n$ are equal, and the nonzero singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$ of P equal the square roots

of those eigenvalues. Also, zero is a singular value of P if either PP^* or P^*P has a zero eigenvalue. Therefore, there are $\max\{m, n\}$ singular values, counting multiplicities. The corresponding orthonormal bases of eigenvectors, $\{\phi_j\}_{j=1}^m \subset \mathbb{R}_M^m$ and $\{\psi_j\}_{j=1}^n \subset \mathbb{R}^n$, are the singular vectors. This gives

$$P\psi_j = \sigma_j \phi_j, \quad P^*\phi_j = \sigma_j \psi_j \quad \text{if } \sigma_j > 0. \tag{1}$$

Note that $\{\phi_j\}_{j=1}^m$ being orthonormal in $\mathbb{R}_M^m$ means

$$(\phi_i, \phi_j)_M = \phi_j^T M \phi_i = \begin{cases} 0, & \text{if } i \neq j, \\ 1, & \text{if } i = j. \end{cases}$$

For more information about the singular value decomposition of operators acting on Hilbert spaces, see, e.g., [22, Chapters VI–VIII], [23, Chapter 30], [24, Sections VI.5–VI.6].

In POD applications, it is typical to only need information about singular vectors corresponding to nonzero singular values. Let $k = \operatorname{rank}(P)$, i.e., P has exactly k positive singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$. Let $V = [\phi_1, \phi_2, \ldots, \phi_k] \in \mathbb{R}^{m \times k}$ be the matrix of the first k orthonormal eigenvectors of PP^*, and let $W = [\psi_1, \psi_2, \ldots, \psi_k] \in \mathbb{R}^{n \times k}$ be the matrix of the first k orthonormal eigenvectors of P^*P. Then (1) gives

$$PW = V\Sigma, \quad P^*V = W\Sigma, \quad \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_k). \tag{2}$$

Since $\{\psi_j\}_{j=1}^k$ is orthonormal in $\mathbb{R}^n$, we have $W^T W = I$. Also, since $\{\phi_j\}_{j=1}^k$ is orthonormal in $\mathbb{R}_M^m$, we have $V^T M V = I$ or $V^* V = I$, where $V^* = V^T M$. Therefore, (2) is equivalent to

$$P = V\Sigma W^T. \tag{3}$$

Since we are primarily interested in the nonzero singular values and corresponding singular vectors, we primarily consider the decomposition (3) for our theoretical results. For the standard matrix case (i.e., without weighted norms), the decomposition (3) is called by various names in the literature and sometimes definitions in different references conflict.[1] In order to potentially avoid confusion, we call this decomposition the *core SVD*.

Definition 1: For a matrix $P : \mathbb{R}^n \to \mathbb{R}^m_M$ with exactly k positive singular values, a *core SVD* of P is given by $P = V\Sigma W^T$, where $V \in \mathbb{R}^{m\times k}$, $\Sigma \in \mathbb{R}^{k\times k}$, and $W \in \mathbb{R}^{n\times k}$ are defined above. ■

Just like the regular SVD, the core SVD is not unique. For example, the columns of V and W can change sign and $P = V\Sigma W^T$ is still a core SVD of P.

Also, we sometimes consider a core SVD of a matrix in the standard sense, i.e., all inner products are unweighted. To be clear, we call this the standard core SVD.

Below, we give some basic properties of the core SVD.

Proposition 1: Suppose $V \in \mathbb{R}^{m\times k}$ has M-orthonormal columns, $W \in \mathbb{R}^{n\times k}$ has orthonormal columns, and $\Sigma \in \mathbb{R}^{k\times k}$ is a positive diagonal matrix with $\Sigma_{11} \geq \Sigma_{22} \geq \cdots \Sigma_{kk} > 0$. If $P : \mathbb{R}^n \to \mathbb{R}^m_M$ satisfies $P = V\Sigma W^T$, then V, Σ, W give a core SVD of P. ■

Proof: First, it is clear that $\text{rank}(P) \leq k$, and therefore P has at most k positive singular values. It is straightforward to check that (2) holds. This implies the k columns of V are eigenvectors of PP^* and the k columns of W are eigenvectors of P^*P, and the corresponding k eigenvalues are the nonzero diagonal entries of Σ. Thus, $P = V\Sigma W^T$ is a core SVD of P. ■

We use the next basic result frequently in this work.

[1] For example, Horn and Johnson [25] call this decomposition the *thin SVD*, but this contradicts with the definition of *thin SVD* in Golub and Van Loan [26]. Furthermore, this definition is called compact SVD in [27]; however, we do not use this terminology to avoid confusion with the SVD of a compact operator.

Proposition 2: Suppose $V_u \in \mathbb{R}^{m\times k}$ has M-orthonormal columns and $W_u \in \mathbb{R}^{n\times k}$ has orthonormal columns. If $Q \in \mathbb{R}^{k\times k}$ has standard core SVD $Q = V_Q \Sigma_Q W_Q^T$ and $P : \mathbb{R}^n \to \mathbb{R}^m_M$ is defined by $P = V_u Q W_u^T$, then

$$P = V\Sigma_Q W^T, \quad V = V_u V_Q, \quad W = W_u W_Q, \tag{4}$$

is a core SVD of P. ■

Proof: First, it is clear P has the representation (4). By the above proposition, we only need to show V has M-orthonormal columns and W also has orthonormal columns. This follows directly since $V_u^T M V_u = I$, $W_u^T W_u = I$, $V_Q^T V_Q = I$, and $W_Q^T W_Q = I$. ■

2.2. COMPUTING THE EXACT SVD WITH RESPECT TO A WEIGHTED INNER PRODUCT

Next, we briefly outline how to compute the exact SVD of a matrix with respect to a weighted inner product using a Cholesky factorization of the weight matrix. In our numerical results in Section 4, we compare the incremental SVD approach in Section 4 to the exact SVD computed using the Cholesky factorization approach discussed here. Note that this Cholesky approach requires storing all of the data, which incremental approaches do not require.

Let $U \in \mathbb{R}^{m\times n}$ be a matrix considered as a mapping $U : \mathbb{R}^n \to \mathbb{R}^m_M$, where $M \in \mathbb{R}^{m\times m}$ is a symmetric positive definite weight matrix. We want to compute the SVD of U as defined in Section 2.1 above.

Let

$$M = R_M^T R_M$$

be the Cholesky factorization of M, where $R_M \in \mathbb{R}^{m\times m}$ is upper triangular and invertible. Transform the matrix U using the Cholesky factor by defining

$$\tilde{U} = R_M U \in \mathbb{R}^{m\times n}.$$

The standard SVD of $\tilde{U}$ (i.e., the SVD with unweighted inner products) gives the SVD of U with respect to the weighted inner product.

Proposition 3: Let $U \in \mathbb{R}^{m\times n}$, and suppose $M \in \mathbb{R}^{m\times m}$ is symmetric positive definite with Cholesky factorization $M = R_M^T R_M$ as above. If $\tilde{U} = \tilde{V}\tilde{\Sigma}\tilde{W}^T$ is the standard core SVD of $\tilde{U} = R_M U \in \mathbb{R}^{m\times n}$, then

$$U = V\Sigma W^T, \quad V = R_M^{-1}\tilde{V}, \quad \Sigma = \tilde{\Sigma}, \quad W = \tilde{W} \tag{5}$$

is the core SVD of $U : \mathbb{R}^n \to \mathbb{R}^m_M$. ■

Proof: We have

$$U = R_M^{-1}\tilde{U} = V\tilde{\Sigma}\tilde{W}^T,$$

and

$$V^*V = V^T M V = \tilde{V}^T R_M^{-T} M R_M^{-1}\tilde{V} = \tilde{V}^T R_M^{-T} R_M^T R_M R_M^{-1}\tilde{V} = \tilde{V}^T\tilde{V} = I.$$

The result follows directly from Proposition 1. ■

Algorithm 1 Exact SVD via Cholesky factorization

Input: $U \in \mathbb{R}^{m\times n}$ and $M \in \mathbb{R}^{m\times m}$ (symmetric positive definite)

1: $R_M = \text{chol}(M)$
2: $\tilde{U} = R_M U$
3: $[\,\tilde{V}, \Sigma, W\,] = \text{svd}(\tilde{U})$
4: Solve for V: $R_M V = \tilde{V}$
5: **return** V, Σ, W

3. BRAND'S INCREMENTAL SVD

Next, we briefly review Brand's incremental SVD algorithm from [21]. The algorithm updates the SVD of a matrix when one or more columns are added to the matrix. A basic implementation of his algorithm has been used for POD computations in [18, 19, 20].

Below, we present the modified version of Brand's incremental SVD algorithm from [20] using single column updates. We first consider the standard inner product, and then a weighted inner product using the Cholesky factorization of the weight matrix. We also briefly discuss why it is beneficial to avoid the Cholesky factorization. Then, in Section 4 below, we propose an extension of Brand's algorithm for a weighted inner product that avoids the Cholesky factorization entirely.

3.1. STANDARD INNER PRODUCT

Suppose we already have the rank-k truncated SVD of a matrix $U \in \mathbb{R}^{m\times n}$ denoted by

$$U = V\Sigma W^T, \tag{6}$$

where $\Sigma \in \mathbb{R}^{k\times k}$ is a diagonal matrix with the k (ordered) singular values of U on the diagonal, $V \in \mathbb{R}^{m\times k}$ is the matrix containing the corresponding k left singular vectors of U, and $W \in \mathbb{R}^{n\times k}$ is the matrix of the corresponding k right singular vectors of U.

Let $c \in \mathbb{R}^m$ be the single column to be added to U. Our goal is to update the above SVD, i.e., we want to find the SVD of $[\, U\, c\,]$. Furthermore, we want to update (Σ, V, W) without forming the matrices U or $[\, U\, c\,]$.

To do this, let $h \in \mathbb{R}^m$ be the projection of c onto the orthogonal complement of the space spanned by the columns of V, i.e.,

$$h = c - VV^T\, c.$$

Also, let p be the magnitude of h and let j be the unit vector in the direction of h, i.e.,

$$p = \|h\|, \quad j = h/p.$$

If $p > 0$, we have the fundamental identity

$$\begin{bmatrix} U & c \end{bmatrix} = \begin{bmatrix} V\Sigma W^T & c \end{bmatrix} = \begin{bmatrix} V & j \end{bmatrix} \begin{bmatrix} \Sigma & V^T c \\ 0 & p \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}^T.$$

As in Proposition 2, we can find the SVD of the updated matrix $[\, U\, c\,]$ by finding the SVD of the middle matrix Q in the right hand side of the above identity. Specifically, if

$$Q := \begin{bmatrix} \Sigma & V^T c \\ 0 & p \end{bmatrix} = V_Q \Sigma_Q W_Q^T$$

is the standard core SVD of Q, then the standard core SVD of $[\, U\, c\,]$ is given by

$$\begin{bmatrix} U & c \end{bmatrix} = \left(\begin{bmatrix} V & j \end{bmatrix} V_Q \right) \Sigma_Q \left(\begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_Q \right)^T. \tag{7}$$

In practice, truncation is performed when p is very small, and also reorthogonalization must be performed on the columns of the updated V and W. We discuss these implementation steps in detail in Section 4.2. We also correct an error in the truncation formulas for the right singular vectors in [20].

Furthermore, as is discussed in [20], we note that only the singular values and left singular vectors need to be computed for many POD applications. However, if one wants to retain an approximation to the data without storing the data, then it is necessary to also compute the right singular vectors.

3.2. WEIGHTED INNER PRODUCT VIA CHOLESKY FACTORIZATION

Next, we discuss incrementally computing the SVD of $[\, U\, c\,]$ with respect to a weighted inner product using a Cholesky factorization of the weight matrix. Specifically, consider $U \in \mathbb{R}^{m\times n}$ as a mapping $U : \mathbb{R}^n \to \mathbb{R}^m_M$, where $M \in \mathbb{R}^{m\times m}$ is symmetric positive definite. Let $c \in \mathbb{R}^m_M$ be the column to be added to U. Our goal is to find the SVD of the updated matrix $[\, U\, c\,]$ considered as a mapping $[\, U\, c\,] : \mathbb{R}^{n+1} \to \mathbb{R}^m_M$.

Suppose we already have the rank-k SVD of U with respect to the weighted inner product given by $U = V\Sigma W^T$. As in Section 2.1, let $M = R_M^T R_M$ be the Cholesky factorization of M, and transform the data: Let

$$\tilde{U} = R_M U = \tilde{V}\Sigma W^T, \quad \tilde{V} = R_M V.$$

Next, we update the standard SVD of $\tilde{U}$ by applying Brand's algorithm as outlined above in Section 3.1 to the transformed updated matrix $R_M[\, U\, c\,] = [\, \tilde{U}\, R_M c\,]$. This gives

$$[\, \tilde{U}\, R_M c\,] = R_M[\, U\, c\,] = \tilde{V}_{\text{new}}\Sigma_{\text{new}}W^T_{\text{new}}. \tag{8}$$

We undo the transformation by multiplying (8) on the left by R_M^{-1}. Therefore, in order to find the updated SVD of $[\, U\, c\,]$ we need to rescale the left singular vectors for $\tilde{U}$ as follows:

$$V_{\text{new}} = R_M^{-1}\tilde{V}_{\text{new}}. \tag{9}$$

This gives the updated SVD: $[\, U\, c\,] = V_{\text{new}}\Sigma_{\text{new}}W^T_{\text{new}}$.

The above approach gives an incremental algorithm for the SVD with respect to a weighted inner product; however, the algorithm has a few drawbacks if m is very large:

- A Cholesky factorization of $M \in \mathbb{R}^{m\times m}$ is required.

- The Cholesky factor R_M of M may not be as sparse as M. (However, it may be possible to avoid a significant loss of sparsity by using ordering methods; see, e.g., [Davis06, Davis16]).

- Solving the linear system $R_M V_{\text{new}} = \tilde{V}_{\text{new}}$ is required.

We avoid all of these drawbacks in the next section by modifying Brand's algorithm to deal with the weighted inner product directly.

4. BRAND'S INCREMENTAL SVD WITH RESPECT TO A WEIGHTED INNER PRODUCT

In this section, we avoid the Cholesky factorization of the weight matrix M and modify Brand's algorithm to treat the weighted inner product case. In the modified algorithm, we do not need to solve any linear systems; we only need to multiply by the weight matrix M. In large-scale applications involving partial differential equations, it is common for M to be sparse; therefore, multiplying by M is a minor computational cost. The modified algorithm has a similar computational cost to the standard algorithm if M is sparse or if multiplying M by a vector can be computed quickly.

We begin in Section 4.1 by describing an idealized version of the incremental SVD algorithm, and we prove it produces the exact core SVD. Then we discuss implementation details in Section 4.2.

4.1. IDEALIZED ALGORITHM WITHOUT TRUNCATION

Suppose we have an exact core SVD of a matrix $U : \mathbb{R}^n \longrightarrow \mathbb{R}^m_M$, and our goal is to update the core SVD when we add a column $c \in \mathbb{R}^m_M$ to U. Furthermore, we want to update the core SVD without forming U or $[\, U\, c\,]$.

First, we propose an idealized version of the algorithm by modifying Brand's algorithm to work in the Hilbert space structure of the weighted inner product space $\mathbb{R}^m_M$ from Section 2. We use Hilbert adjoint operators of matrices, and also the weighted norm $\|x\|_M = x^T M x$ for $x \in \mathbb{R}^m_M$.

The idealized algorithm is given in the following theorem. Again, we present implementation details in Section 4.2 below.

Theorem 1: Let $U : \mathbb{R}^n \longrightarrow \mathbb{R}^m_M$, and suppose $U = V\Sigma W^T$ is the exact core SVD of U, where $V^T M V = I$ for $V \in \mathbb{R}^{m\times k}$, $W^T W = I$ for $W \in \mathbb{R}^{n\times k}$, and $\Sigma \in \mathbb{R}^{k\times k}$. Let $c \in \mathbb{R}^m_M$ and define

$$h = c - VV^* c, \quad p = \|h\|_M, \quad Q = \begin{bmatrix} \Sigma & V^* c \\ 0 & p \end{bmatrix},$$

where $V^* = V^T M$. If $p > 0$ and the standard core SVD of $Q \in \mathbb{R}^{k+1\times k+1}$ is given by

$$Q = V_Q \, \Sigma_Q \, W_Q^T, \tag{10}$$

then the core SVD of $[\, U \, c \,] : \mathbb{R}^{n+1} \longrightarrow \mathbb{R}^m_M$ is given by

$$[\, U \, c \,] = V_u \Sigma_Q W_u^T,$$

where

$$V_u = [\, V \; j \,] \, V_Q, \quad j = h/p, \quad W_u = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_Q.$$

■

Proof: Since $j = h/p = (c - VV^*c)/p$, we have $c = VV^*c + jp$. This gives

$$\begin{aligned}
[\,U\ c\,] &= [\,V\Sigma W^T\ c\,] \\
&= [\,V\Sigma W^T\ \ VV^*c + jp\,] \\
&= [\,V\ j\,] \begin{bmatrix} \Sigma W^T & V^*c \\ 0 & p \end{bmatrix} \\
&= [\,V\ j\,] \begin{bmatrix} \Sigma & V^*c \\ 0 & p \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}^T .
\end{aligned}$$

Next, note

$$[\,V\ j\,]^T M [\,V\ j\,] = \begin{bmatrix} V^TMV & V^TMj \\ (V^TMj)^T & j^TMj \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix}$$

since $V^TMV = V^*V = I$ by assumption,

$$V^TMj = V^*j = V^*(c - VV^*c)/p = (V^*c - V^*c)/p = 0,$$

and

$$j^TMj = \frac{\|h\|_M^2}{p^2} = \frac{\|h\|_M^2}{\|h\|_M^2} = 1.$$

Also, since $W^TW = I$,

$$\begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}^T \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix}.$$

Proposition 2 gives the result. ■

4.2. ALGORITHM DETAILS: INITIALIZATION, TRUNCATION, AND ORTHOG-ONALIZATION

In Section 4.1, we demonstrated an idealized approach to computing the SVD with respect to a weighted inner product incrementally, i.e., by adding one column at a time. Next, we give implementation details concerning initialization, truncation, and orthogonalization.

Initialization. We initialize the SVD with a single column of data c by setting

$$\Sigma = \| c \|_M = (c^T M c)^{1/2}, \quad V = c\Sigma^{-1}, \quad W = 1.$$

We note that although the matrix M may be positive definite in theory, small round off errors may cause $c^T M c$ to be very small and negative in practice. Therefore, throughout this work, when computing the weighted norm we use absolute values under the square root. For example, we actually set $\Sigma = (|c^T M c|)^{1/2}$. We also note that c should be nonzero to initialize. The procedure is given in Algorithm 2.

Algorithm 2 Initialize incremental SVD with respect to weighted inner product

Input: $c \in \mathbb{R}^{m\times 1}$, $c \neq 0$, $M \in \mathbb{R}^{m\times m}$ (symmetric positive definite)

1: $\Sigma = (|c^T M c|)^{1/2}$
2: $V = c\Sigma^{-1}$
3: $W = 1$
4: **return** V, Σ, W

Truncation part 1. The exact SVD update result in Theorem 3 requires $p = \|c - VV^* c\|_M > 0$. When p is small enough, i.e., $p <$ tol for a given tolerance tol, we extend the truncation update approach of Brand [21] to the current weighted norm framework.

If $p < \text{tol}$, we approximate and set $p = 0$. Since $p = \|c - VV^*c\|_M$, this implies $c = VV^*c$. This gives

$$\begin{aligned}
[\,U\ c\,] &= [\,V\Sigma W^T\ c\,] \\
&= [\,V\Sigma W^T\ VV^*c\,] \\
&= [\,V\ 0\,] \begin{bmatrix} \Sigma W^T & V^*c \\ 0 & 0 \end{bmatrix} \\
&= [\,V\ 0\,] \begin{bmatrix} \Sigma & V^*c \\ 0 & 0 \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}^T.
\end{aligned}$$

Similarly to Section 4.1, define $Q \in \mathbb{R}^{k+1\times k+1}$ by

$$Q = \begin{bmatrix} \Sigma & V^*c \\ 0 & 0 \end{bmatrix}.$$

If the full standard SVD of Q is given by $Q = V_Q\Sigma_Q W_Q^T$, where $V_Q, \Sigma_Q, W_Q \in \mathbb{R}^{k+1\times k+1}$, then

$$\begin{aligned}
Q &= V_Q \begin{bmatrix} \Sigma_{Q_{(1:k,1:k)}} & 0 \\ 0 & 0 \end{bmatrix} W_Q^T \\
&= V_Q \begin{bmatrix} \Sigma_{Q_{(1:k,1:k)}} \left(W_{Q_{(:,1:k)}}\right)^T \\ 0 \end{bmatrix} \\
&= V_{Q_{(:,1:k)}} \Sigma_{Q_{(1:k,1:k)}} \left(W_{Q_{(:,1:k)}}\right)^T.
\end{aligned}$$

This gives

$$
\begin{aligned}
[\,U\ c\,] &= [\,V\ 0\,]Q \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}^T \\
&= [\,V\ 0\,]V_{Q_{(:,1:k)}}\Sigma_{Q_{(1:k,1:k)}}\left(W_{Q_{(:,1:k)}}\right)^T \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}^T \\
&= VV_{Q_{(1:k,1:k)}}\Sigma_{Q_{(1:k,1:k)}}\left(\begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_{Q_{(:,1:k)}}\right)^T .
\end{aligned}
$$

This suggests the following update

$$
V \longrightarrow VV_{Q_{(1:k,1:k)}}, \quad \Sigma \longrightarrow \Sigma_{Q_{(1:k,1:k)}}, \quad W \longrightarrow \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_{Q_{(:,1:k)}}.
$$

We note that the rank of the SVD is not increased even though we added a column. Furthermore, the formula for the update of W given here corrects an error in [21] (the matrix $[W, 0; 0, 1]$ is missing from the update formula).

Orthogonalization. In the idealized algorithm in Section 4.1, the SVD update yields orthonormal left and right singular vectors. However, in practice, small numerical errors cause a loss of orthogonality. Following [20], we reorthogonalize when the weighted inner product between the first and last left singular vectors is greater than some tolerance. Specifically, we apply a modified M-weighted Gram-Schmidt procedure with reorthogonalization to the columns of V; the columns of the resulting matrix are orthonormal with respect to the M-weighted inner product. See Algorithm 3, which is a modification for the weighted inner product of the Gram-Schmidt code in [28, Algorithm 6.11, page 307, Section 6.5.6].

Algorithm 3 Modified M-weighted Gram-Schmidt with reorthogonalization

Input: $V \in \mathbb{R}^{m\times r}$, $M \in \mathbb{R}^{m\times m}$

$Q = V$
for $k = 1$ to m **do**
  for $i = 1$ to $k - 1$ **do**
    for $t = 1$ to 2 (reorthogonalize) **do**
      $E = Q(:, i)^T M Q(:, k)$
      $Q(:, k) = Q(:, k) - E Q(:, i)$
      $R(i, k) = R(i, k) + E$
    end for
  end for
  $R(k, k) = \text{sqrt}(Q(:, k)^T M Q(:, k))$
  $Q(:, k) = Q(:, k)/R(k, k)$
end for
return Q

Truncation part 2. The orthogonalization step described above is a large part of the computational cost of the incremental SVD algorithm. If the incremental SVD update is to be repeated for a large number of added columns, the number of nonzero singular values can increase quickly and the computational cost of the orthogonalization steps will be large. In such a case, it is important to keep only the singular values of interest to the application. Usually, singular values very near zero (and their corresponding singular vectors) are not required for POD applications. Therefore, during an incremental SVD update, we keep only the singular values (and corresponding singular vectors) above a user specified tolerance.

Complete Implementation. Our implementation of the incremental SVD update algorithm for a weighted inner product is given in Algorithm 4. Our implementation is modeled after the algorithm in [20] (without a weighted inner product). As is noted in [20], for many POD applications only the singular values and left singular vectors need to be updated and stored. However, if one desires to be able to approximately reconstruct the entire dataset (without storing the data), then one must update and store the singular values and both left and right singular vectors.

Algorithm 4 Incremental SVD with weighted inner product

Input: $V \in \mathbb{R}^{m\times k}$, $\Sigma \in \mathbb{R}^{k\times k}$, $W \in \mathbb{R}^{n\times k}$, $c \in \mathbb{R}^{m}$, $M \in \mathbb{R}^{m\times m}$, tol, $\mathrm{tol}_{\mathrm{sv}}$

% Prepare for SVD update

$d = V^T M c$, $p = \mathrm{sqrt}(|(c - Vd)^T M (c - Vd)|)$

if $(p < \mathrm{tol})$ **then**

  $Q = \begin{bmatrix} \Sigma & d \\ 0 & 0 \end{bmatrix}$

else

  $Q = \begin{bmatrix} \Sigma & d \\ 0 & p \end{bmatrix}$

end if

$[\, V_Q, \Sigma_Q, W_Q \,] = \mathrm{svd}(Q)$

% SVD update

if ($p < \mathrm{tol}$) or ($k \geq m$) **then**

  $V = V V_{Q_{(1:k,1:k)}}$, $\Sigma = \Sigma_{Q_{(1:k,1:k)}}$, $W = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_{Q_{(:,1:k)}}$

else

  $j = (c - Vd)/p$

  $V = [V\ j] V_Q$, $\Sigma = \Sigma_Q$, $W = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_Q$

  $k = k + 1$

end if

% Orthogonalize if necessary

if ($|V^T_{(:,\mathrm{end})} M V_{(:,1)}| > \min(\mathrm{tol}, \mathrm{tol} \times m)$) **then**

  $V = \mathrm{modifiedGSweighted}(V, M)$ % Algorithm 3

end if

% Neglect small singular values: truncation

if $(\Sigma_{(r,r)} > \mathrm{tol}_{\mathrm{sv}})$ and $(\Sigma_{(r+1,r+1)} < \mathrm{tol}_{\mathrm{sv}})$ **then**

  $\Sigma = \Sigma_{(1:r,1:r)}, \quad V = V_{(:,1:r)}, \quad W = W_{(:,1:r)}$

end if

return V, Σ, W

5. INCREMENTAL POD FOR TIME VARYING FUNCTIONS

POD is often used to extract mode shapes or basis functions from solutions of time dependent PDEs. In this case, the dataset consists of a time varying function taking values in a Hilbert space X with inner product $(\cdot, \cdot)$ and corresponding norm $\|\cdot\|$. More information about POD for this type of data can be found in, e.g., [29, 30, 11, 12].

In this section, we show how to compute the POD of the data in this setting using the modified incremental SVD algorithm proposed in Section 4. We focus on approximating continuous POD in Section 5.1, and then consider data expanded in basis functions for X in Section 5.2. We also briefly consider data generated by a finite difference method in Section 5.3.

5.1. APPROXIMATE CONTINUOUS POD

Let w be in $L^2(0,T;X)$, i.e., roughly, $\int_0^T \|w(t)\|^2\,dt < \infty$. Define the continuous POD operator $Z : L^2(0,T) \to X$ by

$$Zg = \int_0^T w(t)\, g(t)\, dt.$$

In practice, we typically only have access to the data at discrete points in time $\{t_j\}_{j=1}^{s+1}$, where $0 \le t_1 < t_2 < \ldots < t_{s+1} \le T$. For $j = 1, \ldots, s$, let $\delta_j = t_{j+1} - t_j$ denote the jth time step. As is well-known (see, e.g., [11, 12]), we rescale the data below by $\delta_j^{1/2}$ in order to arrive at a discrete POD operator.

Approximate the time integral in the definition of the POD operator Z by a Riemann sum with left hand endpoint:

$$\begin{aligned} Zg &\approx \sum_{j=1}^{s} w(t_j)\, \delta_j\, g(t_j) \\ &= \sum_{j=1}^{s} u_j\, f_j, \quad u_j = \delta_j^{1/2} w(t_j), \quad f_j = \delta_j^{1/2} g(t_j). \end{aligned}$$

Therefore, define the (discrete) POD operator $K : \mathbb{R}^s \to X$ by

$$Kf = \sum_{j=1}^{s} u_j f_j, \quad f = [f_1, \ldots, f_s]^T.$$

5.2. DATA EXPANDED IN BASIS FUNCTIONS

In POD applications, a common way to collect data from a time dependent PDE is via numerical simulation. Many approximate solution methods for PDEs (such as finite element and discontinuous Galerkin methods) are Galerkin-type, i.e., the approximate solution data is expressed in terms of a basis of a finite dimensional subspace of X. Let $\{\phi_k\}_{k=1}^m \subset X$ be a collection of linearly independent basis functions, and assume the rescaled data $u_j = \delta_j^{1/2} w(t_j) \in X$ can be expressed as

$$u_j = \delta_j^{1/2} w(t_j) = \sum_{k=1}^{m} U_{k,j} \phi_k, \quad \text{for } j = 1, \ldots, s.$$

In Appendix 1, we show that the SVD of $K : \mathbb{R}^s \to X$ can be computed using the SVD of the coefficient data matrix $U : \mathbb{R}^s \to \mathbb{R}^m$, where the weight matrix $M \in \mathbb{R}^{m \times m}$ has entries $M_{j,k} = (\phi_j, \phi_k)$. We leave the precise statement of the result to the appendix, but we note that if $\{\sigma_i, f_i, c_i\}$ are the nonzero singular values and corresponding singular vectors of $U : \mathbb{R}^s \to \mathbb{R}^m_M$, then $\{\sigma_i, f_i, x_i\}$ are the nonzero singular values and corresponding singular vectors of K, where

$$x_i = \sum_{k=1}^{m} c_{i,k} \phi_k$$

and $c_{i,k}$ is the kth entry of the vector c_i.

Let U_j denote the jth column of the coefficient data matrix U. Algorithm 5 gives the incremental POD algorithm for time varying data. As mentioned previously, for many POD applications only the singular values and left singular vectors (the POD modes) need to be updated and stored. However, if one also updates and stores the right singular vectors

then it is possible to approximately reconstruct the entire dataset without storing the data. Furthermore, we note that the POD eigenvalues are the squares of the POD singular values.

Algorithm 5 Time varying incremental POD

Input: $\{\delta_j\}$, $\{U_j\}$, $M \in \mathbb{R}^{m\times m}$, tol, tol_{sv} % δ_j, U_j, M as described above
1: $[V, \Sigma, W] = \text{initializeSVD}(U_1, M)$ % Algorithm 2
2: **for** $j = 2, \ldots, s$ **do**
3: $\quad [V, \Sigma, W] = \text{incrementalSVD}(V, \Sigma, W, U_j, \text{tol}, \text{tol}_{\text{sv}}, M)$ % Algorithm 4
4: **end for**
5: $W = \text{diag}(\delta)^{-1/2} W$ % undo rescaling, $\delta = [\delta_1, ..., \delta_s]$
6: **return** V, Σ, W

Remark 1: Many researchers remove the average of the data, and then compute the POD of this new data. Such a computation can be performed incrementally without storing the data. We give a brief overview of the algorithm with a weighted inner product in Appendix 2. ■

In some of our numerical results, we compare the time varying incremental SVD with the exact time varying SVD computed using the Cholesky factorization of M as in Section 2.2. We must modify Algorithm 1 for the exact SVD to account for the rescaling by the square roots of the time steps. Let $D = \text{diag}(\delta)$, where $\delta = [\delta_1, ..., \delta_s]$. The modified exact SVD algorithm is shown in Algorithm 6. Again, note that this exact algorithm requires storing all of the data, which incremental algorithms do not require.

Algorithm 6 Exact time varying SVD via Cholesky factorization

Input: $U = [U_1, \ldots, U_s]$, M, D % U_j, M, D as described above
1: $R_M = \text{chol}(M)$
2: $\tilde{U} = R_M U$
3: $[\, \tilde{V}, \Sigma, \tilde{W}\,] = \text{svd}(\tilde{U})$
4: Solve for V: $R_M V = \tilde{V}$
5: $W = D^{-1/2}\tilde{W}$
6: **return** V, Σ, W

5.3. DATA FROM A FINITE DIFFERENCE METHOD

Although we focus on Galerkin-type simulation methods for PDEs in this work, we briefly consider incremental POD for data generated by one type of numerical method for PDEs that is not of Galerkin-type: finite difference methods. The key is to focus on the Hilbert space inner product and its approximation.

Suppose a finite difference method is used to approximate the solution of a scalar time dependent PDE on a bounded domain $\Omega \subset \mathbb{R}^d$, and the goal is to approximate the POD of the data with respect to the $L^2(\Omega)$ inner product. For a function u, let $u_f \in \mathbb{R}^m$ denote the vector of approximations to the function u evaluated at the m finite difference nodes. Then

$$(u, v)_{L^2(\Omega)} = \int_\Omega u(x)\, v(x)\, dx \approx \sum_{i=1}^{m} \eta_i u_{f,i} v_{f,i} = v_f^T M u_f,$$

where $\{\eta_i\}_{i=1}^m$ are positive quadrature weights and $M = \mathrm{diag}(\eta_1, \ldots, \eta_m)$. It is possible to apply the modified incremental SVD algorithm in this work with the weight matrix M. However in this case the data can be rescaled by the square roots of the quadrature weights and Brand's incremental SVD algorithm can be used without a weighted inner product. Once the POD modes are computed, the modes must be multiplied by the diagonal matrix $M^{-1/2}$ so that they are orthonormal with respect to the M weighted inner product.

If instead the goal is to approximate the POD of the data with respect to the $H^1(\Omega)$ inner product, then we have

$$(u, v)_{H^1(\Omega)} = \int_\Omega u\, v + \nabla u \cdot \nabla v\, dx \approx v_f^T M u_f.$$

Here, the matrix M is obtained by approximating the integral using quadrature with positive weights and approximating the gradients by finite difference approximations. In this case, the weight matrix M is not diagonal and so the rescaling idea described above is not applicable; however, the incremental SVD algorithm with weight matrix M can be applied.

6. NUMERICAL RESULTS

In this section, we present numerical results for the incremental POD algorithm applied to time varying finite element solution data for two PDEs: (i) a 1D Burgers' equation, and (ii) a 2D Navier-Stokes equation. The first problem serves as a small test problem with varying time steps. For the second problem, we consider fixed time steps and both small-scale and large-scale computations. For the small-scale problems, we stored all of the simulation data to compare the standard SVD (computed using Algorithm 6) with the incremental SVD (computed using Algorithm 5). For the large-scale problem, we did not store the simulation data and only computed the incremental SVD using Algorithm 5).

For all of the examples reported here, we used the standard L^2 inner product for the POD computations. This corresponds to the matrix SVD with respect to a weighted inner product, where the weight matrix M is the standard finite element mass matrix. For the 1D Burgers' equation example, we also tested the incremental POD using the standard H^1 inner product, which yields a different matrix M. We do not report these results here; we found the algorithm performance is similar in this case to the L^2 performance. We also tested the incremental approach to POD for the removed average data for the 1D Burgers' equation example (as outlined in Appendix 2). Again, we found that the performance of the algorithm is similar to the other cases, and so we do not report the results here.

6.1. EXAMPLE 1: 1D BURGERS' EQUATION

We begin with a small test problem. Consider 1D Burgers' equation with zero Dirichlet boundary conditions

$$\frac{\partial w}{\partial t}(t,x) + w(t,x)\,\frac{\partial w}{\partial x}(t,x) = \frac{1}{Re}\frac{\partial^2 w}{\partial x^2}(t,x), \quad -1 < x < 1.$$

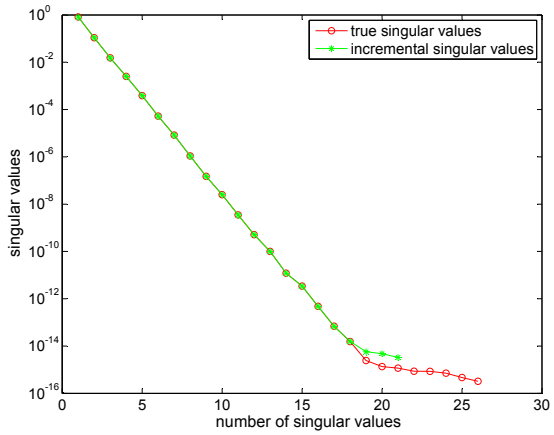


Figure 1. Example 1, $Re = 20$: Exact versus incremental singular values

We used piecewise linear finite elements with 1000 equally spaced notes to approximate the solution of this PDE with $Re = 20$ and initial condition $w(0, x) = \sin(\pi x)$. We used Matlab's `ode23s` to approximate the solution of the resulting nonlinear ODE system on the time interval $0 \leq t \leq 2$. The solver returned the approximate solution at 26 points in time in that interval; the time steps were not equally spaced. At each time point, the finite element coefficient vector had length 998.

For the incremental POD algorithm, we set tol $= 10^{-14}$ and $\text{tol}_{\text{sv}} = 10^{-15}$. Recall, the first tolerance tol is the truncation tolerance for the incremental algorithm, while the second tolerance tol_{sv} is the truncation tolerance for the singular values. In this example and in the examples below, we set tol_{sv} very small in order to test the accuracy of the very small singular values and the corresponding singular vectors; in practice, a very small singular value tolerance is likely rarely needed.

Figure 1 shows the exact versus the incrementally computed singular values. We see excellent agreement for all singular values down to near the singular value tolerance (10^{-14}). Note that the incremental SVD algorithm only returns 21 singular values due to the singular value truncation. A few of the exact and incrementally computed right singular vectors and POD modes are shown in Figures 2 and 3. Again, we see excellent agreement.

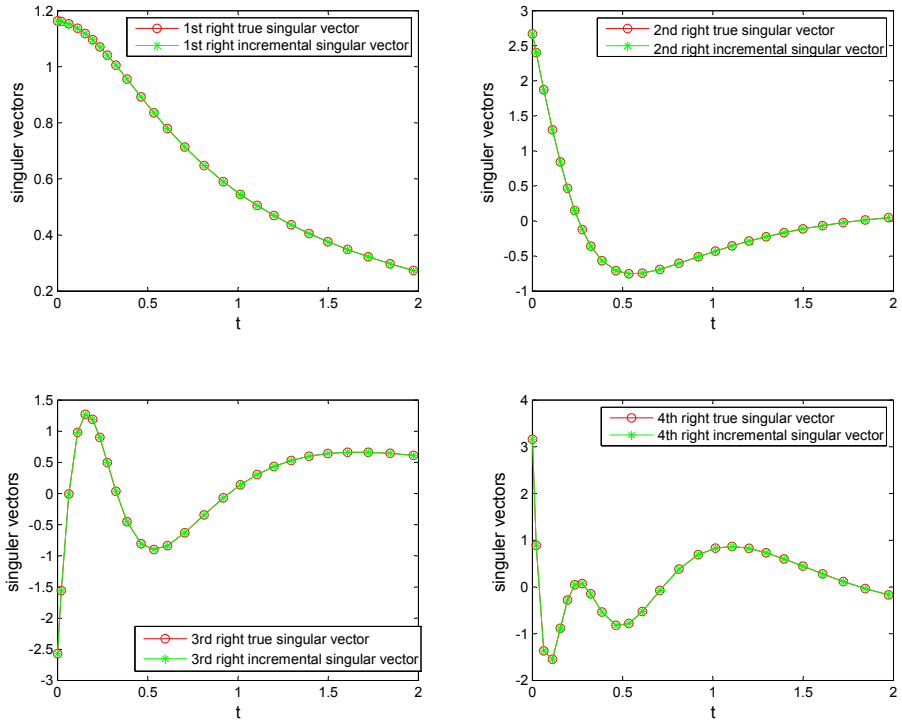


Figure 2. Example 1, $Re = 20$: Exact versus incremental right singular vectors

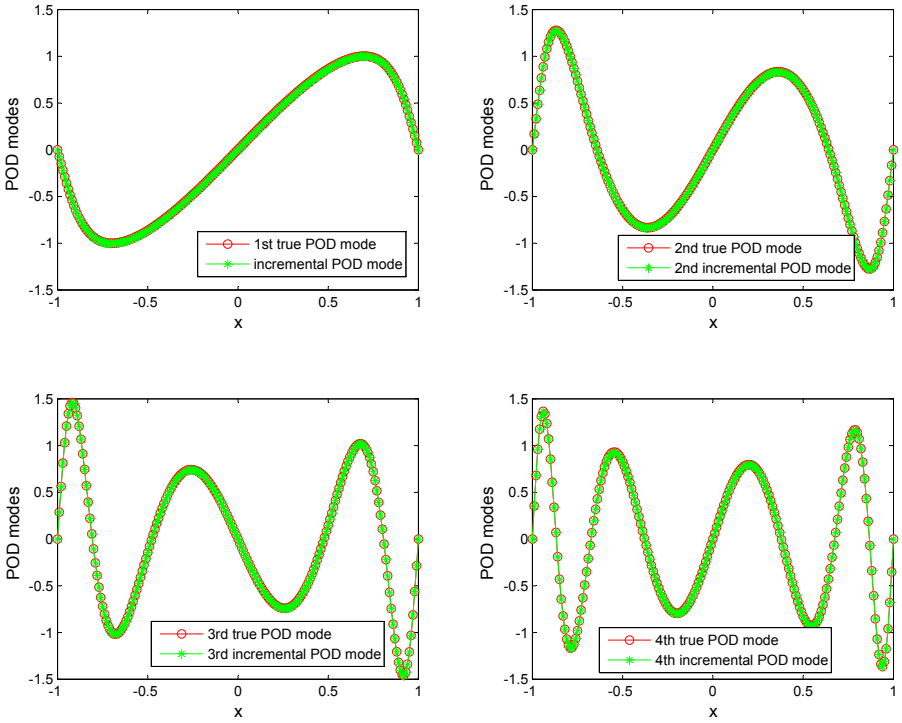


Figure 3. Example 1, $Re = 20$: Exact versus incremental POD modes

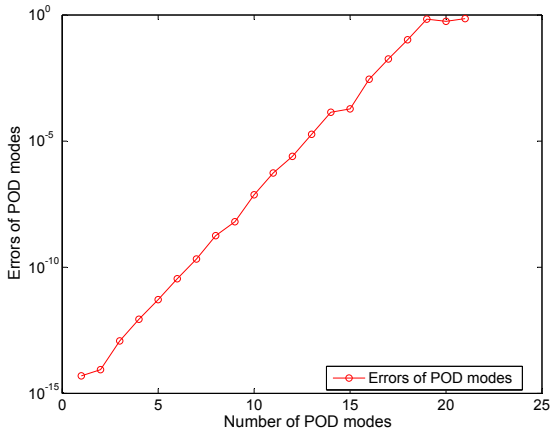


Figure 4. Example 1, $Re = 20$: Errors between true and incremental POD modes

Next, Figure 4 shows the weighted norm error between the exact and incrementally computed POD modes. The errors for the dominant POD modes (corresponding to the largest singular values) are extremely small. The errors in the POD modes increase slowly and monotonically as the corresponding singular values approach zero. The number of highly accurate POD modes is quite large; the first 12 modes are computed to an accuracy level of at least 10^{-5}. The 12th singular value is $O(10^{-9})$. In many POD applications, POD modes are not required for POD singular values that are this small. (Recall, the POD eigenvalues are the squares of the POD singular values.) The incremental POD algorithm works very well for this problem.

6.2. EXAMPLE 2: 2D NAVIER-STOKES EQUATION

For our second example, we consider a 2D laminar flow around a cylinder with circular cross-section [31]. The flow is governed by the time dependent incompressible Navier-Stokes equations with Reynolds number $Re = 100$, and we consider a rectangular spatial domain of length 2.2 and width 0.41. The diameter of the cylinder is 0.1, and it is centered at the point $(0.2, 0.2)$. For the initial condition, we take the steady state solution of the same problem with Reynolds number 40 (instead of 100). On the right boundary of the rectangle (the outlet), we consider stress free boundary conditions. The boundary

conditions on all other walls are Dirichlet boundary conditions. The Dirichlet velocity data on the left wall of the rectangle (the inlet) is $(\frac{6y(0.41-y)}{0.41^2}, 0)$. The Dirichlet data on all other boundaries is zero.

The primary goal of this example is to test the incremental POD algorithm on a problem with more complex solution behavior than the first example (the 1D Burgers' equation). First, we use a coarse grid and a relatively small number of time steps over a short time interval in order to compute the exact errors compared to the exact SVD (with respect to the weighted inner product). We do not attempt to simulate over a longer time period in order to obtain similar numerical results to POD works in the literature (see, e.g., [32, 33]).

For the simulation, we consider the time interval $0 \leq t \leq 1$ and time step 0.01. The finite element mesh is generated by Triangle [34, 35] with local refinement near the cylinder; also, the mesh is polygonal and only approximately fits the circular boundary of the cylinder. We used standard Taylor-Hood elements, and backward Euler for the time stepping for simplicity.

We first consider a coarse mesh. At each time point, the velocity finite element coefficients are vectors of length 55552. We have 101 total solution snapshots. For the incremental SVD computation, we take tol $= 10^{-10}$ and the singular value tolerance $\text{tol}_{\text{sv}} = 10^{-12}$.

The incremental SVD algorithm returns 33 singular values and corresponding singular vectors. Figure 5 shows the exact versus the incremental singular values. We see excellent agreement for all singular values down to near the singular value tolerance (10^{-10}). The first four exact and incrementally computed right singular vectors are shown in Figure 6, and the agreement is again excellent. Figure 7 shows the horizontal and vertical components of the 1st, 5th, and 10th velocity POD modes.

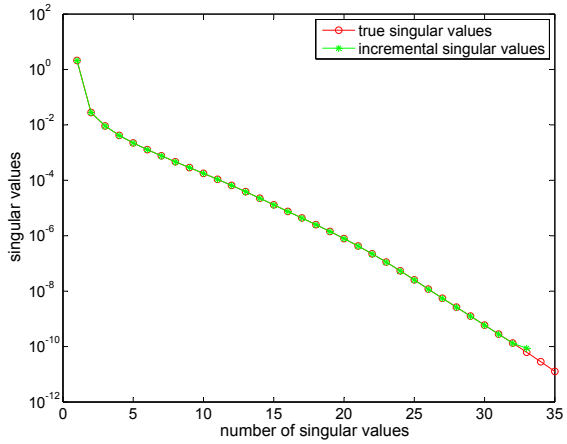


Figure 5. Example 2, $Re = 100$: Exact versus incremental singular values

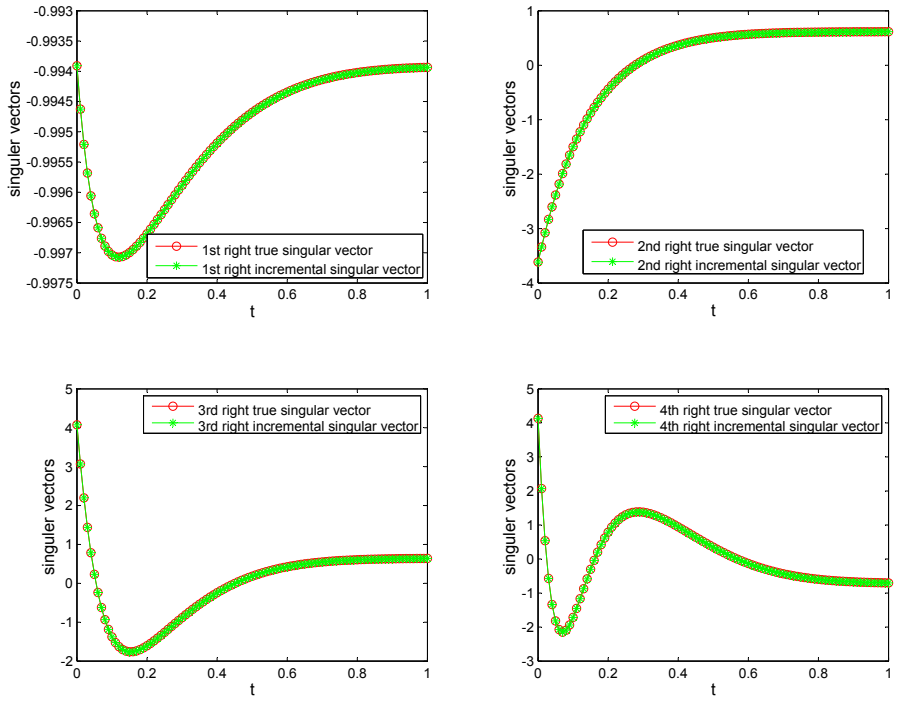


Figure 6. Example 2, $Re = 100$: Exact versus incremental right singular vectors

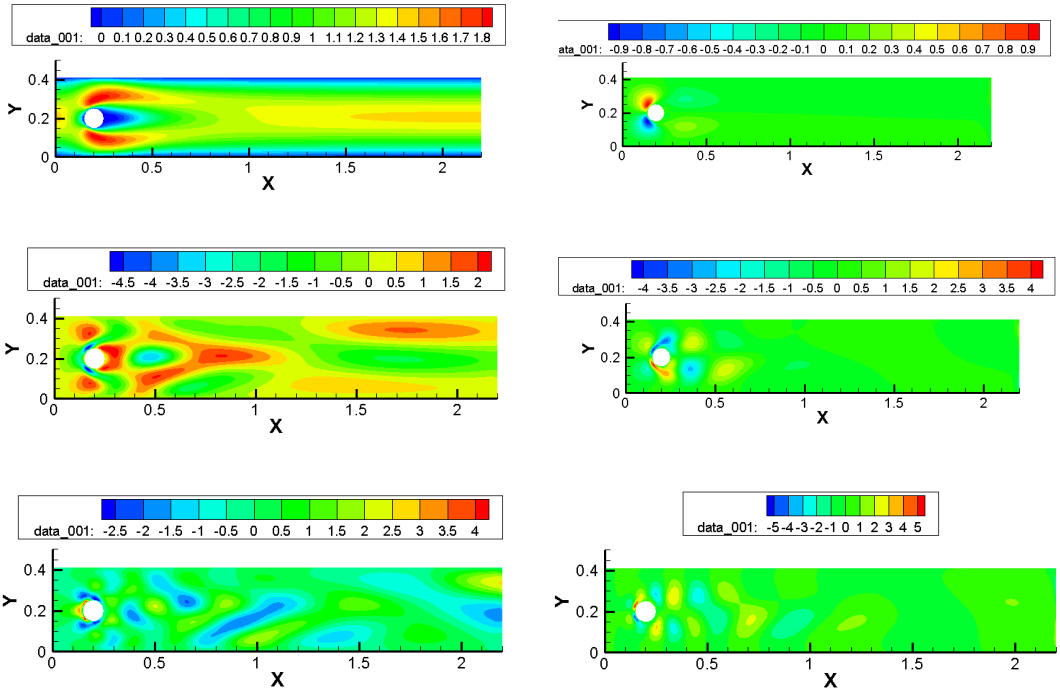


Figure 7. Example 2, *Re* = 100: 1st, 5th, and 10th incremental velocity POD modes (from top to bottom); horizontal components are on the left, and vertical components are on the right

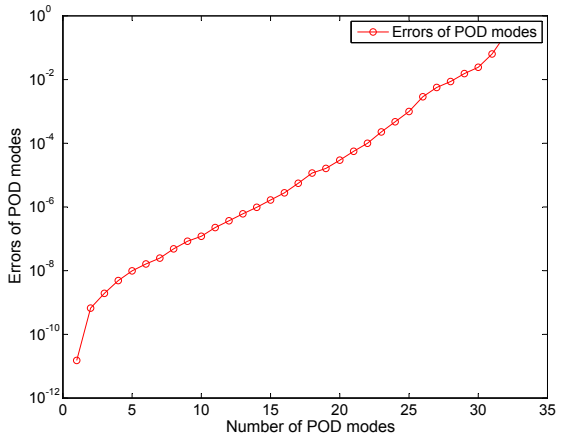


Figure 8. Example 2, $Re = 100$: Errors between true and incremental POD modes

Figure 8 shows the weighted norm error between the exact and incrementally computed velocity POD modes. The error behaves in a similar fashion to the POD mode error in the first example (Figure 4). Again, the errors for the dominant POD modes are extremely small, and the errors increase slowly and monotonically as the corresponding singular values approach zero. Furthermore, there are a large number of highly accurate POD modes. Again, the incremental POD algorithm is very accurate for this problem.

We also tested the same problem with a smaller time step of 0.001 (instead of 0.01); this gives 1001 solution snapshots. We also reduced the algorithm tolerance to tol $= 10^{-12}$. Using the same the singular value tolerance $\text{tol}_{\text{sv}} = 10^{-12}$, the algorithm returned 97 singular values and corresponding singular vectors. We again found that the incremental approach gave accurate results (not shown).

Next, we return to the larger time step 0.01, but now use a fine mesh for the finite element discretization. Each of the 101 flow velocity snapshots has a finite element coefficient vector of length nearly 2 million (1978904). In this case, we did not store the solution data or compute the exact SVD; we only performed the POD computations incrementally. Also, we compared the incremental SVD to the incremental SVD computed on the coarse finite element mesh (with the same time step). We found both incremental

SVD computations gave similar singular values and singular vectors (not shown), as we would expect from POD theory. Specifically, the finite element solution should converge to the solution of the PDE as the mesh is refined; therefore, the POD eigenvalues and modes must converge (see, e.g., [30, 11, 12].

7. CONCLUSION

We extended Brand's incremental SVD algorithm [21] to treat data expanded in basis functions from a Hilbert space. Many numerical methods for PDEs generate data of this form. Specifically, we reformulated Brand's matrix algorithm in a weighted norm setting using functional analytic techniques. We proved that an idealized version of the algorithm exactly updates the SVD when a new column is added to the data. We also considered time varying data by incorporating the quadrature on the time integral into the incremental approach.

We used the left singular vectors to compute the POD modes for the collected data. Standard methods for computing the POD modes require storing the whole large dataset; in contrast, using an incremental SVD algorithm only requires storing one snapshot of the data at a time. Therefore, the incremental approach drastically reduces the memory requirement for computing the POD of the data. Furthermore, the computational cost of the incremental approach is also much lower than standard approaches. Moreover, by truncating small singular values (and corresponding singular vectors) during the incremental update, we reduce the computational cost of orthgonalizing the stored singular vectors.

We tested our approach on finite element simulation data with the L^2 inner product for a 1D Burgers' equation and a 2D Navier-Stokes equation. For the small-scale computational cases, we compared the incremental SVD results with the exact SVD and found excellent agreement. We also found that the incremental algorithm worked very well using a different inner product and also if we removed the average from the data (again with an incremental approach without storing the data). We also tested the algorithm on a fine

mesh for the Navier-Stokes problem with nearly 2 million velocity unknowns. We did not consider an optimized or parallel implementation of the algorithm in this work; this would be of interest to explore in the future. Although we showed the proposed algorithm is exact in an idealized case, we did not perform an error analysis of the algorithm with truncation in this work. In our numerical experiments, we found that the results were not sensitive to the truncation tolerances, as long as the tolerances were chosen relatively small (such as 10^{-8} and smaller). An analysis may provide more insight into the accuracy of the algorithm with truncation and the choices of the tolerances; we leave this to be explored elsewhere. Although we showed the proposed algorithm is exact in an idealized case, we did not perform an error analysis of the algorithm with truncation in this work. In our numerical experiments, we found that we obtained very accurate results for many choices of the truncation tolerances, as long as the tolerances were chosen relatively small (such as 10^{-8} and smaller). An analysis may provide more insight into the accuracy of the algorithm with truncation and the choices of the tolerances; we leave this for future work.

APPENDIX

1. POD IN A HILBERT SPACE AND THE MATRIX SVD WITH A WEIGHTED NORM

Let X be a Hilbert space with inner product $(\cdot, \cdot)$, and suppose $\{u_j\}_{j=1}^s \subset X$. Define the POD operator $K : \mathbb{R}^s \to X$ by

$$Kf = \sum_{j=1}^{s} u_j f_j, \quad f = [f_1, \ldots, f_s]^T. \tag{11}$$

The Hilbert adjoint operator $K^* : X \to \mathbb{R}^s$ satisfies $(Kf, x) = (f, K^*x)_{\mathbb{R}^s}$ for all $f \in \mathbb{R}^s$ and $x \in X$. It can be checked that

$$K^*x = \left[(x, u_1), (x, u_2), \ldots, (x, u_s)\right]^T. \tag{12}$$

Since K has rank at most s, K is compact and has a singular value decomposition. Let $\{\sigma_i, f_i, x_i\}$ be the core singular values and singular vectors of K, i.e., the nonzero singular values and corresponding singular vectors of K. Then

$$Kf_i = \sigma_i x_i, \tag{13}$$

$$K^* x_i = \sigma_i f_i. \tag{14}$$

In the proposition below, we consider the case where each u_j is expressed in terms of a finite set of basis functions. We show that the core singular values and singular vectors of K can be computed by finding the core SVD of a coefficient matrix with respect to a weighted inner product.

Proposition 4: Suppose $\{\phi_k\}_{k=1}^m \subset X$ are linearly independent, and assume $u_j \in X$ is given by

$$u_j = \sum_{k=1}^{m} U_{k,j}\phi_k, \quad \text{for } j = 1, \ldots, s. \tag{15}$$ ■

Let the matrices $M \in \mathbb{R}^{m\times m}$ and $U \in \mathbb{R}^{m\times s}$ have entries $M_{j,k} := (\phi_j, \phi_k)$ and $U_{k,l}$, for $j, k = 1, \ldots, m$ and $l = 1, \ldots, s$. Then $\{\sigma_i, f_i, c_i\} \subset \mathbb{R}\times\mathbb{R}^s\times\mathbb{R}^m_M$ are the core singular values and singular vectors of $U : \mathbb{R}^s \to \mathbb{R}^m_M$ if and only if $\{\sigma_i, f_i, x_i\} \subset \mathbb{R} \times \mathbb{R}^s \times X$ are the core singular values and singular vectors of $K : \mathbb{R}^s \to X$, where c_i and x_i are related by

$$x_i = \sum_{k=1}^{m} c_{i,k}\phi_k \quad \text{for all } i.$$

Proof: First, since $\{\phi_k\}_{k=1}^m \subset X$ is a linearly independent set, we know M is symmetric positive definite.

Next, assume $Kf_i = \sigma_i x_i$ (13) is satisfied with $\sigma_i > 0$. Substitute in the expansion for u_j (15) and use the definition of K in (11) to obtain

$$\sum_{j=1}^{s} \sum_{k=1}^{m} U_{k,j} f_{i,j} \phi_k = \sigma_i x_i, \tag{16}$$

where $f_{i,j}$ denotes the jth entry of the vector f_i. Therefore,

$$x_i = \sum_{l=1}^{m} c_{i,l} \phi_l, \quad c_{i,l} = \frac{1}{\sigma_i} \sum_{j=1}^{s} f_{i,j} U_{l,j}. \tag{17}$$

Let $c_i \in \mathbb{R}_M^m$ denote the vector with entries $c_{i,k}$. Then we have

$$Uf_i = \sigma_i c_i \quad \text{for all } i. \tag{18}$$

Note the above argument is reversible, i.e., if we assume $Uf_i = \sigma_i c_i$ with $\sigma_i > 0$ as in (18), then we obtain $Kf_i = \sigma_i x_i$, where x_i is defined in (17).

Next, we proceed similarly with $K^* x_i = \sigma_i f_i$ (14) and $\sigma_i > 0$. Using the definition of K^* in (12), the expansion for u_j in (18), and the expansion for x_i in (17) gives

$$\begin{aligned}
\sigma_i f_i &= \sum_{l=1}^{m} \sum_{k=1}^{m} \left[\left(c_{i,l} \phi_l, U_{k,1} \phi_k \right), \ldots, \left(c_{i,l} \phi_l, U_{k,s} \phi_k \right) \right]^T \\
&= \sum_{l=1}^{m} \sum_{k=1}^{m} \left[c_{i,l} M_{l,k} U_{k,1}, \ldots, c_{i,l} M_{l,k} U_{k,s} \right]^T \\
&= \left[c_i^T M U_1, \ldots, c_i^T M U_s \right]^T \\
&= \left(c_i^T M U \right)^T,
\end{aligned}$$

where U_j denotes the jth column of the matrix U. Since $U^* = U^T M$, we have

$$U^* c_i = \sigma_i f_i \quad \text{for all } i. \tag{19}$$

Again, this argument is reversible, i.e., $U^* c_i = \sigma_i f_i$ in (19) with $\sigma_i > 0$ implies $K^* x_i = \sigma_i f_i$, where x_i is defined in (17).

Therefore, we have

$$U f_i = \sigma_i c_i, \quad U^* c_i = \sigma_i f_i \quad \text{for all } i$$

if and only if

$$K f_i = \sigma_i x_i, \quad K^* x_i = \sigma_i f_i \quad \text{for all } i.$$

Next, suppose $\{\sigma_i, f_i, x_i\} \subset \mathbb{R} \times \mathbb{R}^s \times X$ are the core singular values and singular vectors of $K : \mathbb{R}^s \to X$. To show $\{\sigma_i, f_i, c_i\} \subset \mathbb{R} \times \mathbb{R}^s \times \mathbb{R}^m_M$ are the core singular values and singular vectors of $U : \mathbb{R}^s \to \mathbb{R}^m_M$, where $c_i = \sigma_i^{-1} U f_i$, we only need to show $\{c_i\} \subset \mathbb{R}^m_M$ is orthonormal. We show this as follows. Using $c_j = \sigma_j^{-1} U f_j$, $U^* c_i = \sigma_i f_i$ (19), and $\{f_i\} \subset \mathbb{R}^s$ is orthonormal gives

$$\begin{aligned}
\left(c_i, c_j\right)_M &= \frac{1}{\sigma_j}\left(c_i, U f_j\right)_M \\
&= \frac{1}{\sigma_j}\left(U^* c_i, f_j\right)_{\mathbb{R}^s} \\
&= \frac{\sigma_i}{\sigma_j}\left(f_i, f_j\right)_{\mathbb{R}^s} \\
&= \frac{\sigma_i}{\sigma_j}\delta_{ij} = \delta_{ij}.
\end{aligned}$$

Therefore, $\{c_i\} \subset \mathbb{R}^m_M$ is orthonormal.

Finally, suppose $\{\sigma_i, f_i, c_i\} \subset \mathbb{R} \times \mathbb{R}^s \times \mathbb{R}^m_M$ are the core singular values and singular vectors of $U : \mathbb{R}^s \to \mathbb{R}^m_M$. To show $\{\sigma_i, f_i, x_i\} \subset \mathbb{R} \times \mathbb{R}^s \times X$ are the core singular values and singular vectors of $K : \mathbb{R}^s \to X$, where x_i is defined in (17), we only need to show $\{x_i\} \subset X$ is orthonormal. This follows directly from $\{c_i\} \subset \mathbb{R}^m_M$ being an orthonormal set:

$$\left(x_i, x_j\right) = c_j^T M c_i = \delta_{ij}.$$

This completes the proof. ■

2. INCREMENTAL SVD AFTER REMOVING THE AVERAGE

Some authors apply POD on the data after removing the average of the data. Such a computation has recently been performed incrementally in [19] by applying an algorithm for the additive modification of an SVD [36]. A similar procedure can be done for time varying data with a weighted norm (as considered in Section 5). We do not give the details of the procedure here; however, we show how Brand's algorithm for the additive modification of the SVD [36] can be extended to the case of a weighted norm.

Theorem 2: Let $M \in \mathbb{R}^{m\times m}$ be symmetric positive definite, and let $a \in \mathbb{R}^m_M$ and $b \in \mathbb{R}^n$. Suppose $U : \mathbb{R}^n \longrightarrow \mathbb{R}^m_M$ has core SVD given by $U = V\Sigma W^T$, where $V^T MV = I$ for $V \in \mathbb{R}^{m\times k}$, $W^T W = I$ for $W \in \mathbb{R}^{n\times k}$, and $\Sigma \in \mathbb{R}^{k\times k}$. Define

$$m = V^* a, \quad p = a - Vm, \quad p_a = \|p\|_M, \tag{20}$$

$$n = W^T b, \quad d = b - Wn, \quad d_b = \|d\|_{\mathbb{R}^n}, \tag{21}$$

where $V^* = V^T M$ and

$$K = \begin{bmatrix} \Sigma + mn^T & d_b m \\ p_a n^T & p_a d_b \end{bmatrix}.$$

If $p_a, d_b > 0$ and the standard core SVD of $K \in \mathbb{R}^{k+1\times k+1}$ is given by

$$K = V_K \Sigma_K W_K^T, \tag{22}$$

then the core SVD of $U + ab^T$ is given by

$$U + ab^T = V_u \Sigma_K W_u^T,$$

where

$$V_u = [\, V\ r\,]\, V_K, \quad r = p_a^{-1} p, \quad W_u = [\, W\ q\,]\, W_K, \quad q = d_b^{-1} d.$$

■

Proof: Rewrite $U + ab^T$ as

$$U + ab^T = V\Sigma W^T + ab^T = [\, V\ a\,] \begin{bmatrix} \Sigma & 0 \\ 0 & 1 \end{bmatrix} [\, W\ b\,]^T. \tag{23}$$

Next, use the definitions in (20) and (21), respectively, to obtain

$$[\, V\ a\,] = [\, V\ r\,] \begin{bmatrix} I & V^* a \\ 0 & p_a \end{bmatrix},$$

$$[\, W\ b\,] = [\, W\ q\,] \begin{bmatrix} I & W^T b \\ 0 & d_b \end{bmatrix}.$$

Substituting these results into (6) gives

$$\begin{aligned} U + ab^T &= [\, V\ r\,] \left(\begin{bmatrix} I & m \\ 0 & p_a \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & n \\ 0 & d_b \end{bmatrix}^T \right) [\, W\ q\,]^T \\ &= [\, V\ r\,] \begin{bmatrix} \Sigma + mn^T & d_b m \\ p_a n^T & p_a d_b \end{bmatrix} [\, W\ q\,]^T. \end{aligned}$$

Next, note

$$[\, V\ r\,]^T M [\, V\ r\,] = \begin{bmatrix} V^T M V & V^T M r \\ (V^T M r)^T & r^T M r \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix}$$

since $V^T M V = V^* V = I$ by assumption,

$$V^T M r = V^* r = V^*(a - Vm)/p_a = (m - m)/p_a = 0,$$

and

$$r^T M r = \frac{\|p\|_M^2}{p_a^2} = \frac{\|p\|_M^2}{\|p\|_M^2} = 1.$$

Also, we have

$$[\, W \; q \,]^T [\, W \; q \,] = \begin{bmatrix} W^T W & W^T q \\ (W^T q)^T & q^T q \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix}$$

since $W^T W = I$,

$$W^T q = W^T(b - Wn)/d_b = (n - n)/d_b = 0,$$

and

$$q^T q = \frac{\|d\|_M^2}{d_b^2} = \frac{\|d\|_M^2}{\|d\|_M^2} = 1.$$

Proposition 2 gives the result. ■

ACKNOWLEDGEMENTS

J. Singler and Y. Zhang were supported in part by National Science Foundation grant DMS-1217122. J. Singler and Y. Zhang thank the Institute for Mathematics and its Applications at the University of Minnesota for funding research visits, during which some of this work was completed. The authors thank the referees for their comments, which helped improve the manuscript.

REFERENCES

[1] D. Amsallem and C. Farhat, "Interpolation method for adapting reduced-order models and application to aeroelasticity," *AIAA Journal*, vol. 46, no. 7, pp. 1803–1813, 2008.

[2] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, Cambridge, second ed., 2012.

[3] H. T. Banks, R. C. H. del Rosario, and R. C. Smith, "Reduced-order model feedback control design: numerical implementation in a thin shell model," *IEEE Trans. Automat. Control*, vol. 45, no. 7, pp. 1312–1324, 2000.

[4] P. Benner, E. Sachs, and S. Volkwein, "Model order reduction for PDE constrained optimization," vol. 165, pp. 303–326, 2014.

[5] M. Gubisch and S. Volkwein, "POD for linear-quadratic optimal control," in *Model Reduction and Approximation: Theory and Algorithms* (P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, eds.), Philadelphia, PA: SIAM, 2017.

[6] M. Gunzburger, N. Jiang, and M. Schneier, "An ensemble-proper orthogonal decomposition method for the nonstationary Navier-Stokes equations," *SIAM J. Numer. Anal.*, vol. 55, no. 1, pp. 286–304, 2017.

[7] R. Ştefănescu, A. Sandu, and I. M. Navon, "POD/DEIM reduced-order strategies for efficient four dimensional variational data assimilation," *J. Comput. Phys.*, vol. 295, pp. 569–595, 2015.

[8] S. Qian, X. Lv, Y. Cao, and F. Shao, "Parameter estimation for a 2D tidal model with POD 4D VAR data assimilation," *Mathematical Problems in Engineering*, vol. 2016, 2016. Article ID 6751537.

[9] L. Sirovich, "Turbulence and the dynamics of coherent structures. I. Coherent structures," *Quarterly of Applied Mathematics*, vol. 45, no. 3, pp. 561–571, 1987.

[10] R. Pinnau, "Model reduction via proper orthogonal decomposition," in *Model order reduction: theory, research aspects and applications*, vol. 13 of *Math. Ind.*, pp. 95–109, Springer, Berlin, 2008.

[11] J. R. Singler, "Convergent snapshot algorithms for infinite-dimensional Lyapunov equations," *IMA J. Numer. Anal.*, vol. 31, no. 4, pp. 1468–1496, 2011.

[12] S. Volkwein, "Proper orthogonal decomposition: Theory and reduced-order modelling (lecture notes)," 2013.

[13] M. Fahl, "Computation of POD basis functions for fluid flows with Lanczos methods," *Math. Comput. Modelling*, vol. 34, no. 1-2, pp. 91–107, 2001.

[14] C. A. Beattie, J. Borggaard, S. Gugercin, and T. Iliescu, "A Domain Decomposition Approach to POD," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 6750–6756, Dec 2006.

[15] Z. Wang, B. McBee, and T. Iliescu, "Approximate partitioned method of snapshots for POD," *J. Comput. Appl. Math.*, vol. 307, pp. 374–384, 2016.

[16] C. Himpe, T. Leibner, and S. Rave, "Hierarchical approximate proper orthogonal decomposition," *arXiv preprint arXiv:1607.05210*, 2016.

[17] C. G. Baker, K. A. Gallivan, and P. Van Dooren, "Low-rank incremental methods for computing dominant singular subspaces," *Linear Algebra Appl.*, vol. 436, no. 8, pp. 2866–2888, 2012.

[18] B. Peherstorfer and K. Willcox, "Dynamic data-driven reduced-order models," *Comput. Methods Appl. Mech. Engrg.*, vol. 291, pp. 21–41, 2015.

[19] M. J. Zahr and C. Farhat, "Progressive construction of a parametric reduced-order model for PDE-constrained optimization," *Internat. J. Numer. Methods Engrg.*, vol. 102, no. 5, pp. 1111–1135, 2015.

[20] G. M. Oxberry, T. Kostova-Vassilevska, W. Arrighi, and K. Chand, "Limited-memory adaptive snapshot selection for proper orthogonal decomposition," *International Journal for Numerical Methods in Engineering*, vol. 109, no. 2, pp. 198–217, 2017.

[21] M. Brand, *Incremental Singular Value Decomposition of Uncertain Data with Missing Values*, pp. 707–720. Computer Vision — ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part I, Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[22] I. Gohberg, S. Goldberg, and M. A. Kaashoek, *Classes of Linear Operators. Vol. I*, vol. 49 of *Operator Theory: Advances and Applications*. Birkhäuser Verlag, Basel, 1990.

[23] P. D. Lax, *Functional Analysis*. Pure and Applied Mathematics (New York), Wiley-Interscience [John Wiley & Sons], New York, 2002.

[24] M. Reed and B. Simon, *Methods of Modern Mathematical Physics I: Functional Analysis*. Academic Press, Inc., New York, second ed., 1980.

[25] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, Cambridge, second ed., 2013.

[26] G. H. Golub and C. F. Van Loan, *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, fourth ed., 2013.

[27] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., *Templates for the solution of algebraic eigenvalue problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[28] W. Gander, M. J. Gander, and F. Kwok, *Scientific computing*, vol. 11 of *Texts in Computational Science and Engineering*. Springer, Cham, 2014.

[29] K. Kunisch and S. Volkwein, "Galerkin proper orthogonal decomposition methods for parabolic problems," *Numer. Math.*, vol. 90, no. 1, pp. 117–148, 2001.

[30] K. Kunisch and S. Volkwein, "Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics," *SIAM J. Numer. Anal.*, vol. 40, no. 2, pp. 492–515, 2002.

[31] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher, *Benchmark Computations of Laminar Flow Around a Cylinder*, pp. 547–566. Wiesbaden: Vieweg+Teubner Verlag, 1996.

[32] I. Akhtar, J. Borggaard, J. A. Burns, H. Imtiaz, and L. Zietsman, "Using functional gains for effective sensor location in flow control: a reduced-order modelling approach," *J. Fluid Mech.*, vol. 781, pp. 622–656, 2015.

[33] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele, "A hierarchy of low-dimensional models for the transient and post-transient cylinder wake," *J. Fluid Mech.*, vol. 497, pp. 335–363, 2003.

[34] J. R. Shewchuk, "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator," in *Applied Computational Geometry: Towards Geometric Engineering* (M. C. Lin and D. Manocha, eds.), vol. 1148 of *Lecture Notes in Computer Science*, pp. 203–222, Springer-Verlag, 1996.

[35] J. R. Shewchuk, "Triangle: A two-dimensional quality mesh generator and delaunay triangulator, version 1.6," 2005.

[36] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra Appl.*, vol. 415, no. 1, pp. 20–30, 2006.

II. ERROR ANALYSIS OF AN INCREMENTAL POD ALGORITHM FOR PDE SIMULATION DATA

Hiba Fareed[1], and John R. Singler[1]

[1] Department of Mathematics and Statistics

Missouri University of Science and Technology

ABSTRACT

In our earlier work [1], we proposed an incremental SVD algorithm with respect to a weighted inner product to compute the proper orthogonal decomposition (POD) of a set of simulation data for a partial differential equation (PDE) without storing the data. In this work, we perform an error analysis of the incremental SVD algorithm. We also modify the algorithm to incrementally update both the SVD and an error bound when a new column of data is added. We show the algorithm produces the exact SVD of an approximate data matrix, and the operator norm error between the approximate and exact data matrices is bounded above by the computed error bound. This error bound also allows us to bound the error in the incrementally computed singular values and singular vectors. We illustrate our analysis with numerical results for three simulation data sets from a 1D FitzHugh-Nagumo PDE system with various choices of the algorithm truncation tolerances.

1. INTRODUCTION

Proper orthogonal decomposition (POD) is a method to find an optimal low order basis to approximate a given set of data. The basis elements are called POD modes, and they are often used to create low order models of high-dimensional systems of ordinary differential equations or partial differential equations (PDEs) that can be simulated easily

and even used for real-time applications. For more about the applications of POD in engineering and applied sciences and POD model order reduction, see, e.g., [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17].

There is a close relationship between the singular value decomposition (SVD) of a set of data and the POD eigenvalues and modes of the data. Due to applications involving functional data and PDEs, many researchers discuss this relationship in weighted inner product spaces and general Hilbert spaces [18, 19, 20, 21]. For the POD calculation, it is important to determine an inner product that is appropriate for the application [22, 23, 24, 25, 7].

Since the size of data sets continues to increase in applications, many researchers have proposed and developed more efficient algorithms for POD computations, the SVD, and other related methods [26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36]. These algorithms have been recently applied in conjunction with techniques such as POD model order reduction and the dynamic mode decomposition, which often consider simulation data from a PDE [37, 25, 38, 39, 40, 41, 42, 43, 44, 45].

In our earlier work [1], we proposed an incremental SVD algorithm for computing POD eigenvalues and modes in a weighted inner product space. Specifically, we considered Galerkin-type PDE simulation data, initialized the SVD on a small amount of the data, and then used an incremental approach to approximately update the SVD with respect to a weighted inner product as new data arrives. The algorithm involves minimal data storage; the PDE simulation data does not need to be stored. The algorithm also involves truncation, and therefore produces approximate POD eigenvalues and modes. We proved the SVD update is exact without truncation.

In this paper, we study the effectiveness of the truncations and deduce error bounds for the SVD approximation. To handle the computational challenge raised by large data sets, we bound the error incrementally. Specifically, we extend the incremental SVD algorithm for a weighted inner product in [1] to compute an error bound incrementally without storing

the data set; see 2, Algorithm 7. We also perform an error analysis in 3 that clarifies the effect of truncation at each step, and provides more insight into the accuracy of the algorithm with truncation and the choices of the two tolerances. We prove the algorithm produces the exact SVD of an approximate data set, and the operator norm error between the exact and approximate data set is bounded above by the incrementally computed error bound. This yields error bounds for the approximate POD eigenvalues and modes. To illustrate the analysis, we present numerical results in 4 for a set of PDE simulation data using various choices of the tolerances. Finally, we present conclusions in 5.

2. BACKGROUND AND ALGORITHM

We begin by setting notation, recalling background material, and discussing the algorithm.

For a matrix $A \in \mathbb{R}^{m\times n}$, let $A_{(p:q,r:s)}$ denote the submatrix of A consisting of the entries of A from rows $p, \ldots, q$ and columns $r, \ldots, s$. Also, if p and q are omitted, then the submatrix should consist of the entries from all rows. A similar convention applies for the columns if r and s are omitted.

Let $M \in \mathbb{R}^{m\times m}$ be symmetric positive definite, and let $\mathbb{R}^m_M$ denote the Hilbert space $\mathbb{R}^m$ with weighted inner product $(x, y)_M = y^T M x$ and corresponding norm $\|x\|_M = (x^T M x)^{1/2}$. For a matrix $P \in \mathbb{R}^{m\times n}$, we can consider P as a linear operator $P : \mathbb{R}^n \to \mathbb{R}^m_M$. In this case, the operator norm of P is

$$\|P\|_{\mathcal{L}(\mathbb{R}^n,\mathbb{R}^m_M)} = \sup_{\|x\|=1} \|Px\|_M.$$

We note that $\mathbb{R}^n$ without a subscript should be understood to have the standard inner product $(x, y) = y^T x$ and Euclidean norm $\|x\| = (x^T x)^{1/2}$. The Hilbert adjoint operator of the matrix $P : \mathbb{R}^n \to \mathbb{R}^m_M$ is the matrix $P^* : \mathbb{R}^m_M \to \mathbb{R}^n$ given by $P^* = P^T M$. We have $(Px, y)_M = (x, P^* y)$ for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m_M$.

In our earlier work [1], we discussed how the proper orthogonal decomposition of a set of PDE simulation data can be reformulated as the SVD of a matrix with respect to a weighted inner product. We do not give the details of the reformulation here, but we do briefly recall the SVD with respect to a weighted inner product since we use this concept throughout this work.

Definition 2: A *core SVD* of a matrix $P : \mathbb{R}^n \to \mathbb{R}^m_M$ is a decomposition $P = V\Sigma W^T$, where $V \in \mathbb{R}^{m\times k}$, $\Sigma \in \mathbb{R}^{k\times k}$, and $W \in \mathbb{R}^{n\times k}$ satisfy

$$V^T MV = I, \quad W^T W = I, \quad \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_k),$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$. The values $\{\sigma_i\}$ are called the (positive) singular values of P and the columns of V and W are called the corresponding singular vectors of P. ■

Since POD applications do not typically require the zero singular values, we do not consider the full SVD of $P : \mathbb{R}^n \to \mathbb{R}^m_M$ in this work. We do note that the SVD of $P : \mathbb{R}^n \to \mathbb{R}^m_M$ is closely related to the eigenvalue decompositions of P^*P and PP^*. See [1, Section 2.1] for more details.

Also, when we consider the SVD (or core SVD) of a matrix without weighted inner products we refer to this as the standard SVD (or standard core SVD).

We consider approximately computing the SVD of a dataset U incrementally by updating the core SVD when each new column c of data is added to the data set. This incremental procedure is performed without forming or storing the original data matrix. Specifically, we focus on the incremental SVD algorithm with a weighted inner product proposed in Algorithm 4 of [1]. The algorithm is based on the following fundamental

identity: if $U = V\Sigma W^T$ is a core SVD, then

$$\begin{aligned}[U\ c\ ] &= [\ V\Sigma W^T\ c\ ] \\ &= [\ V\ j\ ] \begin{bmatrix} \Sigma & V^*c \\ 0 & p \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}^T,\end{aligned}$$

where $j = (c - VV^*c)/p$ and $p = \|c - VV^*c\|_M$ [1]. The algorithm is a modified version of Brand's incremental SVD algorithm [26] to directly treat the weighted inner product. Brand's incremental SVD algorithm without a weighted inner product has been used for POD computations in [42, 45], and our implementation strategy follows the algorithm in [45].

Below, we consider a slight modification of the algorithm from [1]; specifically, we update the algorithm to include a computable error bound e. We show in this work that the algorithm produces the exact core SVD of a matrix $\tilde{U}$ such that $\|U - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^s, \mathbb{R}^m_M)} \leq e$, where U is the true data matrix. This error bound gives information about the approximation error for the singular values and singular vectors; see 3.2 for details.

We take the first step in the incremental SVD algorithm by initializing the SVD and the error bound with a single column $c \neq 0$ as follows:

$$\Sigma = \|\ c\ \|_M = (|c^T M c|)^{1/2}, \quad V = c\Sigma^{-1}, \quad W = 1, \quad e = 0.$$

Here, the error bound e is set to zero since the initial SVD is exact. Also, as mentioned in [1], even though M is positive definite it is possible for round off errors to cause $c^T Mc$ to be very small and negative; we use the absolute value here and throughout the algorithm to avoid this issue.

Then we incrementally update the SVD and the error bound by applying Algorithm 7 when a new column is added. Most of the algorithm is taken directly from [1, Algorithm 4]; we refer to that work for a detailed discussion of the algorithm and details about the implementation.

We note the following:

- The input is an existing SVD V, Σ, and W, a new column c, the weight matrix M, two positive tolerances, and an error bound e.
- Lines 10, 15, 18, 21, and 26 are new, and are simple computations used to update the error bound e.
- In the SVD update stage (lines 1–16), e_p is the error due to p-truncation in line 3.
- In the singular value truncation stage (lines 17–22), e_{sv} is the error due to the singular value truncation in line 19.
- In the orthogonalization stage (lines 23–25), a modified Gram-Schmidt algorithm with reorthogonalization is used; see Section 4.2 in [1].
- The output is the updated SVD and error bound.
- The columns of V are the M-orthonormal POD modes, and the squares of the singular values are the POD eigenvalues.
- If only the POD eigenvalues and modes are required, then the computations involving W can be skipped; however, W is needed if an approximate reconstruction of the entire data set is desired.
- As new columns continue to be added, a user can monitor the computed error bound and lower the tolerances if desired.

Algorithm 7 Incremental SVD and error bound with weighted inner product

Input: $V \in \mathbb{R}^{m\times k}$, $\Sigma \in \mathbb{R}^{k\times k}$, $W \in \mathbb{R}^{n\times k}$, $c \in \mathbb{R}^{m}$, $M \in \mathbb{R}^{m\times m}$, tol, tol_{sv}, e

% Prepare for SVD update

$d = V^T Mc$, $p = \text{sqrt}(|(c - Vd)^T M(c - Vd)|)$

if $(p < \text{tol})$ **then**

$Q = \begin{bmatrix} \Sigma & d \\ 0 & 0 \end{bmatrix}$

else

$Q = \begin{bmatrix} \Sigma & d \\ 0 & p \end{bmatrix}$

end if

$[\, V_Q, \Sigma_Q, W_Q \,] = \text{svd}(Q)$

% SVD update

if ($p < \text{tol}$) or ($k \geq m$) **then**

$V = VV_{Q(1:k,1:k)}$, $\Sigma = \Sigma_{Q(1:k,1:k)}$, $W = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_{Q(:,1:k)}$

$e_p = p$

else

$j = (c - Vd)/p$

$V = [V\ j]V_Q$, $\Sigma = \Sigma_Q$, $W = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_Q$

$k = k + 1$

$e_p = 0$

end if

% Neglect small singular values: truncation

if $(\Sigma_{(r,r)} > \text{tol}_{\text{sv}})$ and $(\Sigma_{(r+1,r+1)} \leq \text{tol}_{\text{sv}})$ **then**

$e_{sv} = \Sigma_{(r+1,r+1)}$

$\Sigma = \Sigma_{(1:r,1:r)}, \quad V = V_{(:,1:r)}, \quad W = W_{(:,1:r)}$

else

$e_{sv} = 0$

end if

% Orthogonalize if necessary

if ($|V^T_{(:,\text{end})} MV_{(:,1)}| > \min(\text{tol}, \text{tol} \times m)$) **then**

$V = \text{modifiedGSweighted}(V, M)$

end if

$e = e + e_p + e_{sv}$

return V, Σ, W, e

3. ERROR ANALYSIS

In this section, we perform an error analysis of Algorithm 7. We show the algorithm produces the exact SVD of another matrix $\tilde{U}$, and bound the error between the matrices.

We assume all computations in the algorithm are performed in exact arithmetic. Therefore, the Gram-Schmidt orthogonalization stage (in lines 23–25) is not considered here. We note that in [1], we considered a Gram-Schmidt procedure with reorthogonalization to minimize the effect of round-off errors; see, e.g., [46, 47, 48, 49]. We leave an analysis of round-off errors in Algorithm 7 to be considered elsewhere.

We begin our analysis in 3.1 by analyzing the error due to each individual truncation step in the algorithm. Then we provide error bounds for the algorithm in 3.2.

3.1. INDIVIDUAL TRUNCATION ERRORS

We begin our analysis of the incremental SVD algorithm by recalling a result from [1]. This result shows that a single column incremental update to the SVD is exact without truncation when $p = \|c - VV^*c\|_M > 0$.

Theorem 3 (Theorem 4.1 in [1]): Let $U : \mathbb{R}^n \longrightarrow \mathbb{R}^m_M$, and suppose $U = V\Sigma W^T$ is an exact core SVD of U, where $V^TMV = I$ for $V \in \mathbb{R}^{m\times k}$, $W^TW = I$ for $W \in \mathbb{R}^{n\times k}$, and $\Sigma \in \mathbb{R}^{k\times k}$. Let $c \in \mathbb{R}^m_M$ and define

$$h = c - VV^*c, \quad p = \|h\|_M, \quad Q = \begin{bmatrix} \Sigma & V^*c \\ 0 & p \end{bmatrix},$$

where $V^* = V^TM$. If $p > 0$ and a standard core SVD of $Q \in \mathbb{R}^{k+1\times k+1}$ is given by

$$Q = V_Q \, \Sigma_Q \, W_Q^T, \tag{1}$$

then a core SVD of $[\, U \;\; c \,] : \mathbb{R}^{n+1} \longrightarrow \mathbb{R}^m_M$ is given by

$$[\, U \;\; c \,] = V_u \Sigma_Q W_u^T,$$

where

$$V_u = [\, V \;\; j \,]\, V_Q, \quad j = h/p, \quad W_u = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_Q.$$

■

Next, we analyze the incremental SVD update in the case when the added column c satisfies $p = \|c - VV^*c\|_M = 0$.

Lemma 1: Let $U = V\Sigma W^T$, c, h, p, and Q be given as in 3, and assume $p = \|c - VV^*c\|_M = 0$. If the full standard SVD of $Q \in \mathbb{R}^{k+1 \times k+1}$ is given by $Q = V_Q \Sigma_Q W_Q^T$, where $V_Q, \Sigma_Q, W_Q \in \mathbb{R}^{k+1 \times k+1}$, then

$$V_Q = \begin{bmatrix} V_{Q(1:k,1:k)} & 0 \\ 0 & 1 \end{bmatrix}, \quad \Sigma_Q = \begin{bmatrix} \Sigma_{Q(1:k,1:k)} & 0 \\ 0 & 0 \end{bmatrix}, \quad \Sigma_{Q(1:k,1:k)} > 0,$$

and a standard core SVD of $R = Q_{(1:k,1:k+1)} = [\, \Sigma \;\; V^*c \,] \in \mathbb{R}^{k \times k+1}$ is given by

$$R = V_{Q(1:k,1:k)} \Sigma_{Q(1:k,1:k)} (W_{Q(:,1:k)})^T.$$

■

Proof: Let $\sigma_{Q_1} \geq \sigma_{Q_2} \geq \cdots \geq \sigma_{Q_{k+1}} \geq 0$ be the singular values of Q so that $\Sigma_Q = \text{diag}(\sigma_{Q_1}, ..., \sigma_{Q_{k+1}})$. Also, let $\{v_{Q_j}\}$ and $\{w_{Q_j}\}$ be the corresponding orthonormal singular vectors in $\mathbb{R}^{k+1}$, so that

$$V_Q = [v_{Q_1}, \ldots, v_{Q_{(k+1)}}], \quad W_Q = [w_{Q_1}, \ldots, w_{Q_{(k+1)}}],$$

with $V_Q^T V_Q = I$ and $W_Q^T W_Q = I$.

First, we show Q has exactly one zero singular value. Since we know

$$Q^T v_{Q_j} = \sigma_{Q_j} w_{Q_j}, \tag{2}$$

$$Q w_{Q_j} = \sigma_{Q_j} v_{Q_j}, \tag{3}$$

for $j = 1, \ldots, k+1$, the number of zero singular values of Q is precisely equal to the dimension of the nullspace of Q^T. Suppose $v = [v_1, \ldots, v_{k+1}]^T \in \mathbb{R}^{k+1}$ satisfies $Q^T v = 0$. Recall $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k) > 0$, and let $d = V^* c = [d_1, \ldots, d_k]^T$. Then $Q^T v = 0$ implies

$$\begin{bmatrix} \sigma_1 v_1 \\ \sigma_2 v_2 \\ \vdots \\ \sigma_k v_k \\ d_1 v_1 + d_2 v_2 + \ldots + d_k v_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}.$$

Since $\sigma_1 \geq \cdots \geq \sigma_k > 0$, we have $v_j = 0$ for $j = 1, \ldots, k$. This implies the nullspace of Q^T is exactly the span of $e_{k+1} = [0, \ldots, 0, 1]^T \in \mathbb{R}^{k+1}$. Therefore, the nullspace is one dimensional and Q has exactly one zero singular value, i.e., $\sigma_{Q_{k+1}} = 0$ and $\sigma_{Q_1} \geq \sigma_{Q_2} \geq \cdots \geq \sigma_{Q_k} > 0$.

Next, $Qw_{Q_j} = \sigma_j v_{Q_j}$ for $j = 1, \ldots, k$ gives

$$\Rightarrow \begin{bmatrix} \sigma_1 w_{Q_{j,1}} + d_1 w_{Q_{j,k+1}} \\ \sigma_2 w_{Q_{j,2}} + d_2 w_{Q_{j,k+1}} \\ \vdots \\ \sigma_k w_{Q_{j,k}} + d_k w_{Q_{j,k+1}} \\ 0 \end{bmatrix} = \begin{bmatrix} \sigma_j v_{Q_{j,1}} \\ \sigma_j v_{Q_{j,2}} \\ \vdots \\ \sigma_j v_{Q_{j,k}} \\ \sigma_j v_{Q_{j,k+1}} \end{bmatrix}.$$

The last equation gives $v_{Q_{j,k+1}} = 0$ since $\sigma_j > 0$ for $j = 1, \ldots, k$. Therefore, for $j = 1, \ldots, k$,

$$v_{Q_j} = [v_{Q_{j,1}}, v_{Q_{j,2}}, \ldots, v_{Q_{j,k}}, 0]^T,$$

and

$$v_{Q_{k+1}} = [0, 0, \ldots, 0, 1]^T.$$

This implies

$$V_Q = \begin{bmatrix} V_{Q_{(1:k,1:k)}} & 0 \\ 0 & 1 \end{bmatrix},$$

and so the SVD decomposition of Q is given by

$$Q = \begin{bmatrix} V_{Q_{(1:k,1:k)}} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Sigma_{Q_{(1:k,1:k)}} & 0 \\ 0 & 0 \end{bmatrix} W_Q^T.$$

This gives $R = Q_{(1:k,1:k+1)} = \check{V}_Q \check{\Sigma}_Q \check{W}_Q^T$, where $\check{V}_Q = V_{Q_{(1:k,1:k)}}$, $\check{\Sigma}_Q = \Sigma_{Q_{(1:k,1:k)}}$, and $\check{W}_Q = W_{Q_{(1:k+1,1:k)}}$. It can be checked that $\check{V}_Q^T \check{V}_Q = I$ and $\check{W}_Q^T \check{W}_Q = I$ since $V_Q^T V_Q = I$ and $W_Q^T W_Q = I$. Therefore, a standard core SVD of $R \in \mathbb{R}^{k \times k+1}$ is given by $R = \check{V}_Q \check{\Sigma}_Q \check{W}_Q^T$.

The following result is nearly identical to Proposition 2.3 in [1]; the proof is also almost identical and is omitted.

Lemma 2 (Proposition 2.3 in [1]): Suppose $V_u \in \mathbb{R}^{m\times k}$ has M-orthonormal columns and $W_u \in \mathbb{R}^{n\times l}$ has orthonormal columns. If $R \in \mathbb{R}^{k\times l}$ has standard core SVD $R = V_R\Sigma_R W_R^T$ and $P : \mathbb{R}^n \to \mathbb{R}^m_M$ is defined by $P = VRW^T$, then

$$P = V_u\Sigma_u W_u^T, \quad V_u = VV_R, \quad \Sigma_u = \Sigma_R, \quad W_u = WW_R, \tag{4}$$

is a core SVD of P. ■

Next, we complete the analysis of the $p = 0$ case:

Proposition 5: Let $U = V\Sigma W^T$, c, h, p, and Q be given as in 3, and assume $p = \|c - VV^*c\|_M = 0$. If the full standard SVD of $Q \in \mathbb{R}^{k+1\times k+1}$ is given by $Q = V_Q\Sigma_Q W_Q^T$, where $V_Q, \Sigma_Q, W_Q \in \mathbb{R}^{k+1\times k+1}$, then a core SVD of $[\,U\ c\,] : \mathbb{R}^{n+1} \to \mathbb{R}^m_M$ is given by

$$[\,U\ c\,] = V_u\Sigma_u W_u^T,$$

where

$$V_u = VV_{Q(1:k,1:k)}, \quad \Sigma_u = \Sigma_{Q(1:k,1:k)}, \quad W_u = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_{Q(:,1:k)}.$$

Proof: Since $p = 0$, we have $c = VV^*c$ and therefore

$$\begin{bmatrix} U & c \end{bmatrix} = \begin{bmatrix} V\Sigma W^T & VV^*c \end{bmatrix} = V \begin{bmatrix} \Sigma & V^*c \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}^T.$$

The result follows from Lemma 1 and Lemma 2 by taking $P = [\,U\ c\,]$ and $R = [\,\Sigma\ V^*c\,]$. ■

Truncation part 1. Next, we analyze the incremental SVD update in the case when the added column c satisfies $p = \|c - VV^*c\|_M < \text{tol}$. In this case, Algorithm 7 does not compute the SVD of $[\,U\ c\,]$. Instead, Algorithm 7 sets $p = 0$ and returns the exact SVD of $\tilde{U} = [\,U\ VV^*c\,]$. The approximation error in the operator norm is given in the next result.

Proposition 6: Let $U : \mathbb{R}^n \longrightarrow \mathbb{R}^m_M$, and suppose $U = V\Sigma W^T$ is a core SVD of U. If $c \in \mathbb{R}^m_M$, $p = \|c - VV^*c\|_M$, and

$$\tilde{U} = [\, U \; VV^*c \,],$$

then

$$\|[\, U \; c \,] - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^{n+1},\mathbb{R}^m_M)} = p.$$

Proof: For $x = [x_1, \ldots, x_{n+1}]^T \in \mathbb{R}^{n+1}$, we have

$$\begin{aligned}
\|[\, U \; c \,] - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^{n+1},\mathbb{R}^m_M)} &= \sup_{\|x\|=1} \left\|[\, 0 \;\; (c - VV^*c) \,]x\right\|_M \\
&= \sup_{\|x\|=1} \|c - VV^*c\|_M \, |x_{n+1}| \\
&= \|c - VV^*c\|_M,
\end{aligned}$$

where the sup is clearly attained by $x = [0, \ldots, 0, 1]^T \in \mathbb{R}^{n+1}$. ■

Truncation part 2. In Algorithm 7, after the SVD update due to an added column the algorithm truncates any singular values that are smaller than a given tolerance, tol_{sv}. For the matrix case with unweighted inner products, the operator norm error caused by this truncation is well-known to equal the first neglected singular value. This result is also true for a compact linear operator mapping between two Hilbert spaces; see, e.g., [50, Chapters VI–VIII], [51, Chapter 30], and [52, Sections VI.5–VI.6] for more information about the SVD for compact operators. This gives the following result:

Proposition 7: Let $U : \mathbb{R}^n \longrightarrow \mathbb{R}^m_M$, and suppose $U = V\Sigma W^T$ is a core SVD of U. For a given $r > 0$, let $\tilde{U}$ be the rank r truncated SVD of U, i.e.,

$$\tilde{U} = V_{(:,1:r)}\Sigma_{(1:r,1:r)}(W_{(:,1:r)})^T.$$

Then

$$\|U - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^n,\mathbb{R}^m_M)} = \Sigma_{(r+1,r+1)}.$$

3.2. ERROR BOUNDS

Next, we fully explain the computed error bound in Algorithm 7. In a typical application of the algorithm, many new columns of data are added and the POD is updated many times. In the following result, we assume we are at the kth step of this procedure and we have an existing error bound. We prove that Algorithm 7 produces a correct update of the error bound.

More specifically, let $k \in \mathbb{N}$, let $U_k, \tilde{U}_k : \mathbb{R}^k \to \mathbb{R}^m_M$, and assume

$$U_k = V_k \Sigma_k W_k^T, \quad \tilde{U}_k = \tilde{V}_k \tilde{\Sigma}_k \tilde{W}_k^T$$

are core SVDs of U and $\tilde{U}$. Let $c_k \in \mathbb{R}^m_M$ and define $U_{k+1} := [U_k \ c_k] : \mathbb{R}^{k+1} \to \mathbb{R}^m_M$. Furthermore, let $\tilde{U}_{k+1} : \mathbb{R}^{k+1} \to \mathbb{R}^m_M$ be the result of one step of the incremental SVD update applied to $\tilde{U}_k$ so that

$$\tilde{U}_{k+1} = \tilde{V}_{k+1} \tilde{\Sigma}_{k+1} \tilde{W}_{k+1}^T.$$

Therefore, we consider the sequence $\{U_k\}$ to be the exact data matrices, and the sequence $\{\tilde{U}_k\}$ to be the result produced (in exact arithmetic) by Algorithm 7.

In exact arithmetic, there are two stages to Algorithm 7. The first stage is the SVD update in lines 1–16. This stage of the algorithm takes $\tilde{U}_k$ and the added column c and produces the update $\hat{U}_{k+1}$. There are two possible results for $\hat{U}_{k+1}$ depending on the value of p in line 1. The second stage is the singular value truncation applied to $\hat{U}_{k+1}$ (lines 17–22), which produces the final update $\tilde{U}_{k+1}$. Again, there are two possible results for $\tilde{U}_{k+1}$, depending on the singular values of $\hat{U}_{k+1}$. We analyze the error bound for each possible outcome of the algorithm in the result below.

Let the positive tolerances tol and tol_{sv} be fixed. Below, we let p_k denote the value p in line 1 of Algorithm 7. We say that p truncation is applied if $p_k < \text{tol}$. We say the singular value truncation is applied if any of the singular values of $\hat{U}_{k+1}$ are less than tol_{sv}.

In this case, we find a value r so that the first r largest singular values of $\hat{U}_{k+1}$ are greater than tol_{sv}, while the remaining singular values are less than or equal to tol_{sv}. We let $\hat{\sigma}_{r+1}$ denote the largest singular value of $\hat{U}_{k+1}$ such that $\hat{\sigma}_{r+1} \leq \text{tol}_{\text{sv}}$.

Theorem 4: If

$$\|U_k - \tilde{U}_k\|_{\mathcal{L}(\mathbb{R}^k, \mathbb{R}^m_M)} \leq e_k, \quad p_k = \|c_k - \tilde{V}_k \tilde{V}_k^* c_k\|_M,$$

then

$$\|U_{k+1} - \tilde{U}_{k+1}\|_{\mathcal{L}(\mathbb{R}^{k+1}, \mathbb{R}^m_M)} \leq e_{k+1},$$

where

$$e_{k+1} = \begin{cases} e_k, & \text{if no truncation is applied,} \\ e_k + p_k, & \text{if only } p \text{ truncation is applied,} \\ e_k + \hat{\sigma}_{r+1}, & \text{if only the singular value truncation is applied,} \\ e_k + p_k + \hat{\sigma}_{r+1}, & \text{if both truncations are applied.} \end{cases}$$

Proof: Stage 1 of Algorithm 7 (lines 1–16) takes $\tilde{U}_k$ and produces $\hat{U}_{k+1}$. If $p_k \geq \text{tol}$, then 3 gives that the core SVD is updated exactly, i.e.,

$$\hat{U}_{k+1} = [\, \tilde{U}_k \;\; c_k \,] \quad \text{if } p_k \geq \text{tol}.$$

Otherwise, if $p_k < \text{tol}$, then Proposition 6 implies

$$\hat{U}_{k+1} = [\, \tilde{U}_k \;\; \tilde{V}\tilde{V}^* c_k \,] \quad \text{if } p_k < \text{tol},$$

and the error is given by

$$\|[\, \tilde{U}_k \;\; c_k \,] - \hat{U}_{k+1}\|_{\mathcal{L}(\mathbb{R}^{k+1}, \mathbb{R}^m_M)} = p_k.$$

Stage 2 of Algorithm 7 (lines 17–22) takes $\hat{U}_{k+1}$ and produces $\tilde{U}_{k+1}$. If all of the singular values of $\hat{U}_{k+1}$ are greater than tol_{sv}, then $\hat{U}_{k+1} = \tilde{U}_{k+1}$ and there is no error in this stage. Otherwise, let $\hat{\sigma}_{r+1}$ denote the largest singular value of $\hat{U}_{k+1}$ such that $\hat{\sigma}_{r+1} \leq \text{tol}_{\text{sv}}$. In this case, $\tilde{U}_{k+1}$ is simply the rth order truncated SVD of $\hat{U}_{k+1}$, and the error is given by 7:

$$\|\tilde{U}_{k+1} - \hat{U}_{k+1}\|_{\mathcal{L}(\mathbb{R}^{k+1}, \mathbb{R}^m_M)} = \hat{\sigma}_{r+1}.$$

Below, for ease of notation, let $\|\cdot\|$ denote the $\mathcal{L}(\mathbb{R}^{k+1}, \mathbb{R}^m_M)$ operator norm. The error between U_{k+1} and $\tilde{U}_{k+1}$ in the operator norm can be bounded as follows:

$$\|U_{k+1} - \tilde{U}_{k+1}\| \leq \|U_{k+1} - [\,\tilde{U}_k \;\; c_k\,]\| + \|[\,\tilde{U}_k \;\; c_k\,] - \hat{U}_{k+1}\| + \|\hat{U}_{k+1} - \tilde{U}_{k+1}\|.$$

As noted above, the second error term is either zero if p truncation is not applied or p_k otherwise. Also, the third error term is either zero if the singular values truncation is not applied or $\hat{\sigma}_{r+1}$ otherwise. For the first term, we have

$$\begin{aligned}
\|U_{k+1} - [\,\tilde{U}_k \;\; c_k\,]\| &= \|[\,U_k \;\; c_k\,] - [\,\tilde{U}_k \;\; c_k\,]\| \\
&= \|(U_k - \tilde{U}_k)\;\, 0\| \\
&= \sup_{\|x\|=1} \left\|[(U_k - \tilde{U}_k)\;\, 0]x\right\|_M \\
&\leq \|U_k - \tilde{U}_k\|_{\mathcal{L}(\mathbb{R}^k, \mathbb{R}^m_M)} \leq e_k.
\end{aligned}$$

This completes the proof. ■

The result above explains the update of the error bound in one step of Algorithm 7. Now we assume the SVD is initialized exactly when $k = 1$, and then the algorithm is applied for a sequence of added columns $\{c_k\} \subset \mathbb{R}^m_M$, for $k = 2, \ldots, s$.

Corollary 1: Let tol and tol_{sv} be fixed positive constants, and let $\{c_k\} \subset \mathbb{R}^m_M$, for $k = 1, \ldots, s$, be the columns of a matrix U. For $k = 1$, assume the SVD $\tilde{U}_1 = \tilde{V}_1 \tilde{\Sigma}_1 \tilde{W}_1^T$ and error bound $e_1 = 0$ are initialized exactly as described in 2. For $k = 1, \ldots, s-1$, let $\tilde{U}_{k+1} = \tilde{V}_{k+1} \tilde{\Sigma}_{k+1} \tilde{W}_{k+1}^T$ and e_{k+1} be the output of Algorithm 7 applied to the input $\tilde{U}_k = \tilde{V}_k \tilde{\Sigma}_k \tilde{W}_k^T$ and e_k. If T_p represents the total number of times p truncation is applied and T_{sv} represents the total number of times the singular value truncation is applied, then

$$\|U - \tilde{V}_s \tilde{\Sigma}_s \tilde{W}_s\|_{\mathcal{L}(\mathbb{R}^s, \mathbb{R}^m_M)} \leq T_p \text{tol} + T_{\text{sv}} \text{tol}_{\text{sv}}.$$

■

Proof: The proof follows immediately from the previous result, using $p_k \leq \text{tol}$ and $\hat{\sigma}_{r+1} \leq \text{tol}_{\text{sv}}$. ■

The error bound in the result above is not as precise as the error bound computed using Algorithm 7 since the tolerances are only upper bounds on the errors in each step. However, this result does provide some insight into the choice of the tolerances for the algorithm. Specifically, in general there is no reason to expect one of T_p or T_{sv} to be significantly larger than the other; therefore, it seems reasonable to choose equal values for the tolerances. Furthermore, for a very large number of added columns, it is possible that T_p and T_{sv} can be large; therefore, small tolerances should be chosen to preserve accuracy.

Algorithm 7 computes an upper bound on the operator norm error between the exact data matrix U and the approximate truncated SVD $\tilde{U} = \tilde{V} \tilde{\Sigma} \tilde{W}^T$ of the data matrix. (The above corollary also provides another upper bound on the error.) This error bound allows us to bound the error in the incrementally computed singular values and singular vectors. Let $\{\sigma_k, v_k, w_k\}_{k \geq 1}$ and $\{\tilde{\sigma}_k, \tilde{v}_k, \tilde{w}_k\}_{k \geq 1}$ denote the ordered singular values and corresponding

orthonormal singular vectors of $U, \tilde{U} : \mathbb{R}^s \to \mathrm{R}_M^m$ in the result below. The following result follows directly from general results about error bounds for singular values and singular vectors of compact linear operators in 6.

Theorem 5: Let $k \geq 1$, and let $\varepsilon > 0$ such that $\|U - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^s, \mathbb{R}_M^m)} \leq \varepsilon$. Then

$$|\sigma_\ell - \tilde{\sigma}_\ell| \leq \varepsilon \quad \text{for all } \ell \geq 1.$$

Also, for $j = 1, \ldots, k$, define

$$\varepsilon_j = j\varepsilon + 2 \sum_{i=1}^{j-1} \left(\varepsilon_i + \sigma_i E_i^{1/2} \right), \quad E_j = 2 \left(1 - \sqrt{\frac{(\sigma_j - 2\varepsilon_j)^2 - \sigma_{j+1}^2}{\sigma_j^2 - \sigma_{j+1}^2}} \right).$$

If the first $k + 1$ singular values of U are distinct and positive, the singular vector pairs $\{\tilde{v}_j, \tilde{w}_j\}_{j=1}^k$ are suitably normalized, and

$$\varepsilon_j \leq \frac{\sigma_j - \sigma_{j+1}}{2} \quad \text{for } j = 1, \ldots, k,$$

then

$$\|v_j - \tilde{v}_j\|_M \leq E_j^{1/2}, \quad \|w_j - \tilde{w}_j\| \leq E_j^{1/2} + 2\sigma_j^{-1}\varepsilon_j, \quad \text{for } j = 1, \ldots, k. \tag{5}$$ ■

This result indicates we should expect accurate approximate singular values and also accurate approximate singular vectors if ε is small and there is not a small gap in the singular values. We note that POD singular values often decay to zero quickly, and therefore we expect to see lower accuracy in the computed POD modes for smaller singular values due to the small gap. The examples in our first work [1] and the new examples below show both of these expected behaviors for the errors in the approximate singular vectors.

4. NUMERICAL RESULTS

We consider the 1D FitzHugh-Nagumo system

$$\frac{\partial v(t,x)}{\partial t} = \mu \frac{\partial^2 v(t,x)}{\partial x^2} - \frac{1}{\mu} w(t,x) + \frac{1}{\mu} f(v) + \frac{c}{\mu}, \quad 0 < x < 1,$$
$$\frac{\partial w(t,x)}{\partial t} = bv(t,x) - \gamma w(t,x) + c, \quad 0 < x < 1,$$

where $f(v) = v(v - 0.1)(1 - v)$, $\mu = 0.015$, $b = 0.5$, $\gamma = 2$, $c = 0.05$, the boundary conditions are $v_x(t, 0) = -50000t^3 e^{-15t}$, $v_x(t, 1) = 0$, and the initial conditions are zero. This example problem was considered in [53], and we used the interpolated coefficient finite element method from that work to discretize the problem in space. For the finite element method we used continuous piecewise linear basis functions with equally spaced nodes, and we used Matlab's `ode23s` to approximate the solution of the resulting nonlinear ODE system on different time intervals.

For the POD computations, we consider the data $z(t, x) = [v(t, x), w(t, x)]$ in the Hilbert space $L^2(0, 1) \times L^2(0, 1)$ with standard inner product. Now we follow the procedure in our first work [1] to arrive at the weighted SVD problem. At each time step, we rescale the approximate solution data by the square root of the time step; see [1, Section 5.1]. We expand the approximate solution in the finite element basis to obtain the weight matrix M as in [1, Section 5.2]. To compute the POD of the approximate solution data, we compute the SVD of the finite element solution coefficient matrix $U : \mathbb{R}^s \to \mathbb{R}^m_M$, where s is the number of time steps (snapshots) and m is two times the number of finite element nodes.

To illustrate our analysis of the incremental SVD algorithm, we consider three examples:

Example 1 5000 finite element nodes and $s = 491$ snapshots in the time interval $[0, 10]$

Example 2 10000 finite element nodes and $s = 710$ snapshots in the time interval $[0, 15]$

Example 3 50000 finite element nodes and $s = 1275$ snapshots in the time interval $[0, 28]$

We consider relatively small values of $m = 2 \times$ nodes and s in order to test the incremental algorithm against exact SVD computations.

Let U denote the finite element solution coefficient matrix, and let $\tilde{U} = \tilde{V}\tilde{\Sigma}\tilde{W}^T$ denote the incrementally computed approximate SVD of $U : \mathbb{R}^s \to \mathbb{R}^m_M$ produced by Algorithm 7. For each example, we choose various tolerances and compute:

$$\text{Rank} = \text{rank}(\tilde{U}), \quad \text{Exact error} = \|U - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^s, \mathbb{R}^m_M)},$$

$$\text{Incr. error bound} = e \text{ computed by Algorithm 7 at the final snapshot.}$$

The exact SVD of $U : \mathbb{R}^s \to \mathbb{R}^m_M$ and the exact error are both computed using a Cholesky factorization of the weight matrix M following Algorithm 1 in [1]. The exact computations are for testing only since they require storing all of the data.

Tables 1–3 display the computed quantities listed above for the three exampleswith various choices of the p truncation tolerance, tol, and the singular value truncation tolerance, tol_{sv}. We set each tolerance to 10^{-8}, 10^{-10}, or 10^{-12}, for a total of nine tests for each example. In all of the tests, the incrementally computed error bound is larger than the exact error and the error bound is small. Also, the tests indicate that there is no benefit from choosing one tolerance different than the other.

Figure 1 shows the exact and incrementally computed POD singular values and also the weighted norm error between the exact and incrementally computed POD modes with tol and tol_{sv} both equal to 10^{-12}. The errors for the POD modes corresponding to the largest singular values are extremely small (approximately 10^{-12}). The errors in the POD modes increase slowly as the corresponding singular values approach zero. There are many accurate POD modes; the first 30 modes are computed to an accuracy level of at least 10^{-5}. The POD singular value and mode errors behaved similarly for other cases.

Table 1. Example 1 – error between true and incremental SVD

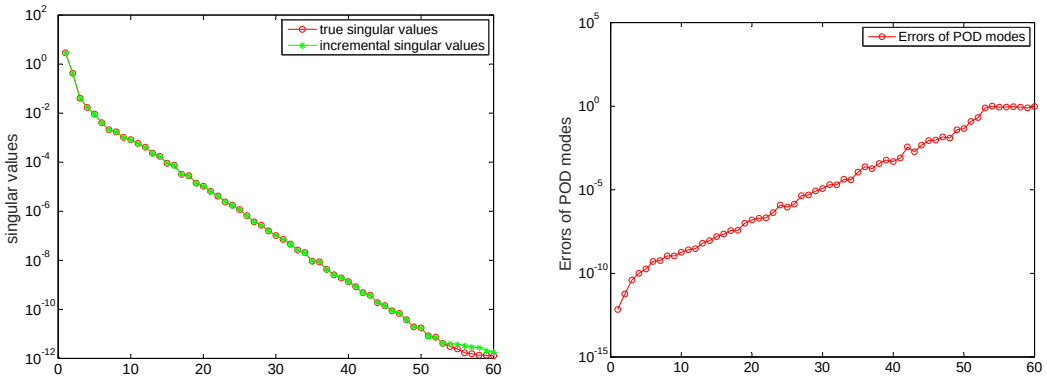
tol	tol_{sv}	Rank	Exact error	Incr. error bound
10^{-8}	10^{-8}	36	$3.6924e-07$	$2.8029e-06$
10^{-8}	10^{-10}	66	$3.1932e-07$	$1.1826e-06$
10^{-8}	10^{-12}	61	$8.5938e-07$	$9.0495e-07$
10^{-10}	10^{-8}	30	$3.9090e-08$	$1.4908e-06$
10^{-10}	10^{-10}	44	$4.4893e-10$	$2.7417e-08$
10^{-10}	10^{-12}	71	$3.9349e-10$	$8.9680e-09$
10^{-12}	10^{-8}	30	$3.9090e-08$	$1.4908e-06$
10^{-12}	10^{-10}	41	$4.5256e-10$	$1.5511e-08$
10^{-12}	10^{-12}	55	$4.4334e-12$	$2.8596e-10$

Table 2. Example 2 – error between true and incremental SVD

tol	tol_{sv}	Rank	Exact error	Incr. error bound
10^{-8}	10^{-8}	35	$3.0859e-07$	$3.6931e-06$
10^{-8}	10^{-10}	66	$1.3881e-07$	$1.1429e-06$
10^{-8}	10^{-12}	64	$3.4657e-07$	$1.5321e-06$
10^{-10}	10^{-8}	31	$4.1497e-08$	$1.7368e-06$
10^{-10}	10^{-10}	45	$5.3142e-10$	$3.6491e-08$
10^{-10}	10^{-12}	74	$7.7348e-10$	$1.1523e-08$
10^{-12}	10^{-8}	30	$4.1497e-08$	$1.7368e-06$
10^{-12}	10^{-10}	41	$4.6086e-10$	$1.8671e-08$
10^{-12}	10^{-12}	59	$4.8658e-12$	$3.4880e-10$

Table 3. Example 3 – error between true and incremental SVD

tol	tol_{sv}	Rank	Exact error	Incr. error bound
10^{-8}	10^{-8}	38	$6.5705e-08$	$4.3271e-06$
10^{-8}	10^{-10}	72	$6.8271e-07$	$1.1523e-06$
10^{-8}	10^{-12}	67	$3.6916e-07$	$2.3847e-06$
10^{-10}	10^{-8}	31	$4.7018e-08$	$2.2388e-06$
10^{-10}	10^{-10}	49	$4.8302e-10$	$4.3655e-08$
10^{-10}	10^{-12}	78	$2.4473e-08$	$2.6825e-08$
10^{-12}	10^{-8}	31	$4.7018e-08$	$2.2388e-06$
10^{-12}	10^{-10}	41	$4.9660e-10$	$2.5022e-08$
10^{-12}	10^{-12}	60	$6.3200e-12$	$5.7438e-10$



(a) POD singular values (b) POD mode errors

Figure 1. Example 3 – exact versus incremental POD computations with tol = $\text{tol}_{\text{sv}} = 10^{-12}$

5. CONCLUSION

In our earlier work [1], we proposed computing the SVD with respect to a weighted inner product incrementally to obtain the POD eigenvalues and modes of a set of PDE simulation data. In this work, we extended the algorithm to update the SVD and an error bound incrementally when a new column is added. We also performed an error analysis of this algorithm by analyzing the error due to each individual truncation. We showed that the algorithm produces the exact SVD of a matrix $\tilde{U}$ such that $\|U - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^s, \mathbb{R}^m_M)} \leq e$, where U is the true data matrix, M is the weight matrix, and e is computed error bound. We also proved error bounds for the incrementally computed singular values and singular vectors. We tested our approach on three example data sets from a 1D FitzHugh-Nagumo PDE system with various choices of the two truncation tolerances. In all of the tests, the incrementally computed error bound was larger than the exact error and the error bound was small. Furthermore, the approximate singular values and dominant singular vectors were accurate. Also, our analysis and the numerical tests suggest that there is no benefit from choosing one algorithm tolerance different than the other.

APPENDIX

Let X and Y be two separable Hilbert spaces, with inner products $(\cdot,\cdot)_X$ and $(\cdot,\cdot)_Y$ and corresponding norms $\|\cdot\|_X$ and $\|\cdot\|_Y$. Below, we drop the subscripts on the inner products and the norms since the space will be clear from the context. Assume $H, H_\varepsilon : X \to Y$ are compact linear operators. In this section, we prove bounds on the error between the singular vectors of H and H_ε assuming the singular values are distinct. Our results rely on techniques from [54, 55].

Let $\{\sigma_k, v_k, w_k\}_{k\geq 1}$ and $\{\sigma_k^\varepsilon, v_k^\varepsilon, w_k^\varepsilon\}_{k\geq 1}$ be the ordered singular values and corresponding orthonormal singular vectors of H and H_ε. They satisfy

$$Hv_k = \sigma_k w_k, \quad H^* w_k = \sigma_k v_k, \quad H_\varepsilon v_k^\varepsilon = \sigma_k^\varepsilon w_k^\varepsilon, \qquad H_\varepsilon^* w_k^\varepsilon = \sigma_k^\varepsilon v_k^\varepsilon, \tag{6}$$

where the star denotes the Hilbert adjoint operator. Also, if $\sigma_k > 0$, then σ_k^2 is the kth ordered eigenvalue of the self-adjoint nonnegative compact operators HH^* and H^*H. First, we recall a well-known bound on the singular values; see, e.g., [56, page 30] and [50, page 99].

Proposition 8: Let $\varepsilon > 0$ such that $\|H - H_\varepsilon\|_{\mathcal{L}(X,Y)} \leq \varepsilon$. Then for all $k \geq 1$ we have

$$|\sigma_k - \sigma_k^\varepsilon| < \varepsilon. \tag{7}$$ ■

In the results below, we require the singular vectors $\{v_k^\varepsilon, w_k^\varepsilon\}$ are suitably normalized. We note that any pair $\{v_k^\varepsilon, w_k^\varepsilon\}$ of singular vectors for a fixed value of k can be rescaled by a constant of unit magnitude and remain a pair of singular vectors. However, due to the relationship 6, we note that both vectors in the pair must be rescaled by the same constant.

The proof of the following result is largely contained in [54, Appendix 2], but we include the proof here to be complete.

Lemma 3: Let $\varepsilon > 0$ such that $\|H - H_\varepsilon\|_{\mathcal{L}(X,Y)} \leq \varepsilon$. If $\sigma_1 > \sigma_2 > 0$, v_1^ε and w_1^ε are suitably normalized, and

$$\varepsilon \leq \frac{\sigma_1 - \sigma_2}{2}, \tag{8}$$

then

$$\|v_1 - v_1^\varepsilon\| \leq E_1^{1/2}, \quad \|w_1 - w_1^\varepsilon\| \leq E_1^{1/2} + 2\sigma_1^{-1}\varepsilon, \quad E_1 = 2\left(1 - \sqrt{\frac{(\sigma_1 - 2\varepsilon)^2 - \sigma_2^2}{\sigma_1^2 - \sigma_2^2}}\right). \tag{9}$$

■

Remark 2: The larger error bound for $\|w_1 - w_1^\varepsilon\|$ is due to the way we assume the singular vectors are normalized in the proof. It is possible to use a different normalization and make the error bound larger for $\|v_1 - v_1^\varepsilon\|$ instead. We comment on the normalization in the proof. ■

Proof: Define $V_1 = \text{span}\{v_1\} \subset X$. We have $X = V_1 \oplus V_1^\perp$, and therefore $v_1^\varepsilon = r_\varepsilon v_1 + x_\varepsilon$ for some constant r_ε and $x_\varepsilon \in X$ satisfies $(x_\varepsilon, v_1) = 0$. This gives $\|x_\varepsilon\|^2 = 1 - |r_\varepsilon|^2$ and also $|r_\varepsilon| \leq 1$. Then

$$\begin{aligned} \|v_1 - v_1^\varepsilon\|^2 &= \|v_1 - r_\varepsilon v_1 - x_\varepsilon\|^2 \\ &= |1 - r_\varepsilon|^2 \|v_1\|^2 + \|x_\varepsilon\|^2 \\ &= 2(1 - \text{Re}(r_\varepsilon)). \end{aligned} \tag{10}$$

Note $\|\sigma_1^\varepsilon w_1^\varepsilon\| = \|H_\varepsilon v_1^\varepsilon\|$ implies

$$\begin{aligned} \sigma_1^\varepsilon &= \|H_\varepsilon v_1^\varepsilon + H v_1^\varepsilon - H v_1^\varepsilon\| \\ &\leq \|H v_1^\varepsilon\| + \|H - H_\varepsilon\| \|v_1^\varepsilon\| \\ &\leq \|H(r_\varepsilon v_1 + x_\varepsilon)\| + \varepsilon \\ &= \|r_\varepsilon \sigma_1 w_1 + H x_\varepsilon\| + \varepsilon. \end{aligned}$$

To estimate this norm, we use $(Hx_\varepsilon, w_1) = (x_\varepsilon, H^* w_1) = \sigma_1 (x_\varepsilon, v_1) = 0$ and also

$$\|Hx_\varepsilon\|^2 = \frac{(H^*Hx_\varepsilon, x_\varepsilon)}{\|x_\varepsilon\|^2}\|x_\varepsilon\|^2 \leq \sup_{x \in V_1^\perp,\, x \neq 0} \frac{(H^*Hx, x)}{\|x\|^2}\|x_\varepsilon\|^2 = \sigma_2^2 \|x_\varepsilon\|^2,$$

where we used the variational characterization of the second eigenvalue σ_2^2 of the self-adjoint compact nonnegative operator H^*H. These results give

$$\begin{aligned} \|r_\varepsilon \sigma_1 w_1 + Hx_\varepsilon\|^2 &= |r_\varepsilon|^2 \sigma_1^2 + \|Hx_\varepsilon\|^2 \\ &\leq |r_\varepsilon|^2 \sigma_1^2 + \sigma_2^2 \|x_\varepsilon\|^2 \\ &= \left(\sigma_1^2 - \sigma_2^2\right)|r_\varepsilon|^2 + \sigma_2^2. \end{aligned}$$

Next, the assumption 8 for ε gives $\varepsilon \leq (\sigma_1 - \sigma_2)/2 \leq \sigma_1/2$, and therefore $\sigma_1 - 2\varepsilon \geq 0$. Also, 7 gives $-\varepsilon \leq \sigma_1^\varepsilon - \sigma_1$, or $\sigma_1^\varepsilon - \varepsilon \geq \sigma_1 - 2\varepsilon \geq 0$. This gives $(\sigma_1^\varepsilon - \varepsilon)^2 \geq (\sigma_1 - 2\varepsilon)^2$, and therefore

$$|r_\varepsilon|^2 \geq \frac{(\sigma_1^\varepsilon - \varepsilon)^2 - \sigma_2^2}{\sigma_1^2 - \sigma_2^2} \geq \frac{(\sigma_1 - 2\varepsilon)^2 - \sigma_2^2}{\sigma_1^2 - \sigma_2^2}.$$

Note that the assumption 8 for ε guarantees that we can take a square root of this estimate.

If v_1^ε is normalized so that r_ε is a nonnegative real number, then 10, $1 - \mathrm{Re}(r_\varepsilon) = 1 - |r_\varepsilon|$, and the above inequality give the desired estimate 9 for $\|v_1 - v_1^\varepsilon\|$. If r_ε is not a nonnegative real number, then rescale the singular vector pair $\{v_1^\varepsilon, w_1^\varepsilon\}$ by $\overline{r}_\varepsilon/|r_\varepsilon|$ to obtain the proper normalization and the bound 9 for $\|v_1 - v_1^\varepsilon\|$.

For w_1 and w_1^ε, it does not appear that we can use a similar proof strategy since we have already rescaled the singular vector pair $\{v_1^\varepsilon, w_1^\varepsilon\}$. Specifically, we can obtain $w_1^\varepsilon = s_\varepsilon w_1 + y_\varepsilon$, but it is not clear that s_ε will be a nonnegative real number and we are unable to rescale again. Therefore, we use $\|H\| = \sigma_1$, $\|H - H_\varepsilon\| \leq \varepsilon$, and $|\sigma_1 - \sigma_1^\varepsilon| \leq \varepsilon$ to

directly estimate:

$$\begin{aligned}\|w_1 - w_1^\varepsilon\| &= \|\sigma_1^{-1} H v_1 - (\sigma_1^\varepsilon)^{-1} H_\varepsilon v_1^\varepsilon\| \\ &\le \|\sigma_1^{-1} H v_1 - \sigma_1^{-1} H v_1^\varepsilon\| + \|\sigma_1^{-1} H v_1^\varepsilon - \sigma_1^{-1} H_\varepsilon v_1^\varepsilon\| + \|\sigma_1^{-1} H_\varepsilon v_1^\varepsilon - (\sigma_1^\varepsilon)^{-1} H_\varepsilon v_1^\varepsilon\| \\ &\le \|v_1 - v_1^\varepsilon\| + \sigma_1^{-1}\varepsilon + |\sigma_1^\varepsilon \sigma_1^{-1} - 1| \\ &\le \|v_1 - v_1^\varepsilon\| + 2\sigma_1^{-1}\varepsilon.\end{aligned}$$

In the result below, note that $\varepsilon_1 = \varepsilon$ and E_1 is defined as in 9 in Lemma 3 above.

Theorem 6: Let $k \ge 1$, and let $\varepsilon > 0$ such that $\|H - H_\varepsilon\|_{\mathcal{L}(X,Y)} \le \varepsilon$. For $j = 1, \ldots, k$, define

$$\varepsilon_j = j\varepsilon + 2\sum_{i=1}^{j-1}\left(\varepsilon_i + \sigma_i E_i^{1/2}\right), \quad E_j = 2\left(1 - \sqrt{\frac{(\sigma_j - 2\varepsilon_j)^2 - \sigma_{j+1}^2}{\sigma_j^2 - \sigma_{j+1}^2}}\right).$$

If the first $k+1$ singular values of H are distinct and positive, the singular vector pairs $\{v_j^\varepsilon, w_j^\varepsilon\}_{j=1}^k$ are suitably normalized, and

$$\varepsilon_j \le \frac{\sigma_j - \sigma_{j+1}}{2} \quad \text{for } j = 1, \ldots, k,$$

then

$$\|v_j - v_j^\varepsilon\| \le E_j^{1/2}, \quad \|w_j - w_j^\varepsilon\| \le E_j^{1/2} + 2\sigma_j^{-1}\varepsilon_j, \quad \text{for } j = 1, \ldots, k. \tag{11}$$ ■

Proof: The proof is by induction. First, the result is true for $k = 1$ by Lemma 3. Next, assume the result is true for all $j = 1, \ldots, k-1$. Define compact linear operators for $j = 2, \ldots, k$ by

$$H^j x = Hx - \sum_{i=1}^{j-1} \sigma_i(x, v_i) w_i, \quad H_\varepsilon^j x = H_\varepsilon x - \sum_{i=1}^{j-1} \sigma_i^\varepsilon(x, v_i^\varepsilon) w_i^\varepsilon,$$

for all $x \in X$. Then the ordered singular values and corresponding singular vectors of H^j and H_ε^j are $\{\sigma_i, v_i, w_i\}_{i\geq j}$ and $\{\sigma_i^\varepsilon, v_i^\varepsilon, w_i^\varepsilon\}_{i\geq j}$.

Note that

$$\begin{aligned}\|H^k x - H_\varepsilon^k x\| &\leq \|(H - H_\varepsilon)x\| + \sum_{i=1}^{k-1} \|\sigma_i^\varepsilon(x, v_i^\varepsilon)w_i^\varepsilon - \sigma_i(x, v_i)w_i\| \\ &\leq \varepsilon\|x\| + \|x\| \sum_{i=1}^{k-1} \left(|\sigma_i^\varepsilon - \sigma_i| + \sigma_i\|v_i^\varepsilon - v_i\| + \sigma_i\|w_i^\varepsilon - w_i\|\right).\end{aligned}$$

Then since the result 4 is true for all $j = 1, \ldots, k-1$, we have $\|H^k - H_\varepsilon^k\| \leq \varepsilon_k$, where

$$\begin{aligned}\varepsilon_k &= \varepsilon + \sum_{i=1}^{k-1} \left(\varepsilon + \sigma_i E_i^{1/2} + \sigma_i\left(E_i^{1/2} + 2\sigma_i^{-1}\varepsilon_i\right)\right) \\ &= k\varepsilon + 2\sum_{i=1}^{k-1} \left(\varepsilon_i + \sigma_i E_i^{1/2}\right).\end{aligned}$$

Applying Lemma 3 to H^k and H_ε^k with $\|H^k - H_\varepsilon^k\| \leq \varepsilon_k$ completes the proof. ■

ACKNOWLEDGEMENTS

The authors thank Mark Opmeer for a helpful discussion.

REFERENCES

[1] H. Fareed, J. Shen, J. R. Singler, and Y. Zhang, "Incremental proper orthogonal decomposition for PDE simulation data," *Computers & Mathematics with Applications*, vol. 75, no. 6, pp. 1942 – 1960, 2018.

[2] T. Colonius and J. Freund, "POD analysis of sound generation by a turbulent jet," in *40th AIAA Aerospace Sciences Meeting & Exhibit*, p. 72, 2002.

[3] R. Zimmermann and S. G?rtz, "Non-linear reduced order models for steady aerodynamics," *Procedia Computer Science*, vol. 1, no. 1, pp. 165 – 174, 2010. ICCS 2010.

[4] R. Zimmermann, "A locally parametrized reduced-order model for the linear frequency domain approach to time-accurate computational fluid dynamics," *SIAM J. Sci. Comput.*, vol. 36, no. 3, pp. B508–B537, 2014.

[5] L. Peng and K. Mohseni, "Nonlinear model reduction via a locally weighted pod method," *International Journal for Numerical Methods in Engineering*, vol. 106, no. 5, pp. 372–396, 2016. nme.5124.

[6] E. A. Christensen, M. Brϕns, and J. N. Sϕrensen, "Evaluation of proper orthogonal decomposition-based decomposition techniques applied to parameter-dependent nonturbulent flows," *SIAM J. Sci. Comput.*, vol. 21, no. 4, pp. 1419–1434, 1999/00.

[7] I. Kalashnikova, M. F. Barone, S. Arunajatesan, and B. G. van Bloemen Waanders, "Construction of energy-stable projection-based reduced order models," *Appl. Math. Comput.*, vol. 249, pp. 569–596, 2014.

[8] D. Amsallem and J. Nordström, "Energy stable model reduction of neurons by nonnegative discrete empirical interpolation," *SIAM J. Sci. Comput.*, vol. 38, no. 2, pp. B297–B326, 2016.

[9] D. N. Daescu and I. M. Navon, "A Dual-Weighted Approach to Order Reduction in 4DVAR Data Assimilation," *Monthly Weather Review*, vol. 136, no. 3, pp. 1026–1041, 2008.

[10] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge Monographs on Mechanics, Cambridge University Press, Cambridge, second ed., 2012.

[11] M. F. Barone, I. Kalashnikova, D. J. Segalman, and H. K. Thornquist, "Stable Galerkin reduced order models for linearized compressible flow," *J. Comput. Phys.*, vol. 228, no. 6, pp. 1932–1946, 2009.

[12] V. M. Calo, Y. Efendiev, J. Galvis, and M. Ghommem, "Multiscale empirical interpolation for solving nonlinear PDEs," *J. Comput. Phys.*, vol. 278, pp. 204–220, 2014.

[13] C. Farhat, T. Chapman, and P. Avery, "Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models," *Internat. J. Numer. Methods Engrg.*, vol. 102, no. 5, pp. 1077–1110, 2015.

[14] X. Xie, D. Wells, Z. Wang, and T. Iliescu, "Numerical analysis of the Leray reduced order model," *J. Comput. Appl. Math.*, vol. 328, pp. 12–29, 2018.

[15] M. Mohebujjaman, L. G. Rebholz, X. Xie, and T. Iliescu, "Energy balance and mass conservation in reduced order models of fluid flows," *J. Comput. Phys.*, vol. 346, pp. 262–277, 2017.

[16] M. Gunzburger, N. Jiang, and M. Schneier, "An ensemble-proper orthogonal decomposition method for the nonstationary Navier-Stokes equations," *SIAM J. Numer. Anal.*, vol. 55, no. 1, pp. 286–304, 2017.

[17] T. Kostova-Vassilevska and G. M. Oxberry, "Model reduction of dynamical systems by proper orthogonal decomposition: error bounds and comparison of methods using snapshots from the solution and the time derivatives," *J. Comput. Appl. Math.*, vol. 330, pp. 553–573, 2018.

[18] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced basis methods for partial differential equations*. Springer, Cham, 2016.

[19] J. R. Singler, "New POD error expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs," *SIAM J. Numer. Anal.*, vol. 52, no. 2, pp. 852–876, 2014.

[20] K. Kunisch and S. Volkwein, "Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics," *SIAM J. Numer. Anal.*, vol. 40, no. 2, pp. 492–515, 2002.

[21] M. Gubisch and S. Volkwein, "POD for linear-quadratic optimal control," in *Model Reduction and Approximation: Theory and Algorithms* (P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, eds.), Philadelphia, PA: SIAM, 2017.

[22] Z. Drmac and A. K. Saibaba, "The discrete empirical interpolation method: Canonical structure and formulation in weighted inner product spaces," *arXiv preprint arXiv:1704.06606*, 2017.

[23] M. Tabandeh, M. Wei, and J. P. Collins, "On the symmetrization in POD-Galerkin model for linearized compressible flows," in *54th AIAA Aerospace Sciences Meeting*, p. 1106, 2016.

[24] G. Serre, P. Lafon, X. Gloerfelt, and C. Bailly, "Reliable reduced-order models for time-dependent linearized Euler equations," *J. Comput. Phys.*, vol. 231, no. 15, pp. 5176–5194, 2012.

[25] D. Amsallem, M. J. Zahr, and K. Washabaugh, "Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction," *Adv. Comput. Math.*, vol. 41, no. 5, pp. 1187–1230, 2015.

[26] M. Brand, *Incremental Singular Value Decomposition of Uncertain Data with Missing Values*, pp. 707–720. Computer Vision — ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part I, Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[27] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra Appl.*, vol. 415, no. 1, pp. 20–30, 2006.

[28] C. G. Baker, K. A. Gallivan, and P. Van Dooren, "Low-rank incremental methods for computing dominant singular subspaces," *Linear Algebra Appl.*, vol. 436, no. 8, pp. 2866–2888, 2012.

[29] Y. Chahlaoui, K. Gallivan, and P. Van Dooren, "Recursive calculation of dominant singular subspaces," *SIAM J. Matrix Anal. Appl.*, vol. 25, no. 2, pp. 445–463, 2003.

[30] M. A. Iwen and B. W. Ong, "A distributed and incremental SVD algorithm for agglomerative data analysis on large networks," *SIAM J. Matrix Anal. Appl.*, vol. 37, no. 4, pp. 1699–1718, 2016.

[31] N. Mastronardi, M. Van Barel, and R. Vandebril, "A note on the recursive calculation of dominant singular subspaces," *Numer. Algorithms*, vol. 38, no. 4, pp. 237–242, 2005.

[32] N. Mastronardi, M. Van Barel, and R. Vandebril, "A fast algorithm for the recursive calculation of dominant singular subspaces," *J. Comput. Appl. Math.*, vol. 218, no. 2, pp. 238–246, 2008.

[33] M. Fahl, "Computation of POD basis functions for fluid flows with Lanczos methods," *Math. Comput. Modelling*, vol. 34, no. 1-2, pp. 91–107, 2001.

[34] C. A. Beattie, J. Borggaard, S. Gugercin, and T. Iliescu, "A Domain Decomposition Approach to POD," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 6750–6756, Dec 2006.

[35] Z. Wang, B. McBee, and T. Iliescu, "Approximate partitioned method of snapshots for POD," *J. Comput. Appl. Math.*, vol. 307, pp. 374–384, 2016.

[36] C. Himpe, T. Leibner, and S. Rave, "Hierarchical approximate proper orthogonal decomposition," *arXiv preprint arXiv:1607.05210*, 2016.

[37] A. Placzek, D.-M. Tran, and R. Ohayon, "A nonlinear POD-Galerkin reduced-order model for compressible flows taking into account rigid body motions," *Comput. Methods Appl. Mech. Engrg.*, vol. 200, no. 49-52, pp. 3497–3514, 2011.

[38] A. Corigliano, M. Dossi, and S. Mariani, "Model order reduction and domain decomposition strategies for the solution of the dynamic elastic-plastic structural problem," *Comput. Methods Appl. Mech. Engrg.*, vol. 290, pp. 127–155, 2015.

[39] B. Peherstorfer and K. Willcox, "Dynamic data-driven reduced-order models," *Comput. Methods Appl. Mech. Engrg.*, vol. 291, pp. 21–41, 2015.

[40] B. Peherstorfer and K. Willcox, "Dynamic data-driven model reduction: adapting reduced models from incomplete data," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 3, p. 11, Mar 2016.

[41] O. T. Schmidt, "An efficient streaming algorithm for spectral proper orthogonal decomposition," *arXiv preprint arXiv:1711.04199*, 2017.

[42] M. J. Zahr and C. Farhat, "Progressive construction of a parametric reduced-order model for PDE-constrained optimization," *Internat. J. Numer. Methods Engrg.*, vol. 102, no. 5, pp. 1111–1135, 2015.

[43] R. Zimmermann, "A closed-form update for orthogonal matrix decompositions under arbitrary rank-one modifications," *arXiv preprint arXiv:1711.08235*, 2017.

[44] R. Zimmermann, B. Peherstorfer, and K. Willcox, "Geometric Subspace Updates with Applications to Online Adaptive Nonlinear Model Reduction," *SIAM J. Matrix Anal. Appl.*, vol. 39, no. 1, pp. 234–261, 2018.

[45] G. M. Oxberry, T. Kostova-Vassilevska, W. Arrighi, and K. Chand, "Limited-memory adaptive snapshot selection for proper orthogonal decomposition," *International Journal for Numerical Methods in Engineering*, vol. 109, no. 2, pp. 198–217, 2017.

[46] L. Giraud and J. Langou, "When modified Gram-Schmidt generates a well-conditioned set of vectors," *IMA J. Numer. Anal.*, vol. 22, no. 4, pp. 521–528, 2002.

[47] L. Giraud, J. Langou, M. Rozloznik, and J. van den Eshof, "Rounding error analysis of the classical Gram-Schmidt orthogonalization process," *Numer. Math.*, vol. 101, no. 1, pp. 87–100, 2005.

[48] L. Giraud, J. Langou, and M. Rozloznik, "The loss of orthogonality in the Gram-Schmidt orthogonalization process," *Comput. Math. Appl.*, vol. 50, no. 7, pp. 1069–1075, 2005.

[49] M. Rozloznik, M. Tuma, A. Smoktunowicz, and J. Kopal, "Numerical stability of orthogonalization methods with a non-standard inner product," *BIT*, vol. 52, no. 4, pp. 1035–1058, 2012.

[50] I. Gohberg, S. Goldberg, and M. A. Kaashoek, *Classes of Linear Operators. Vol. I*, vol. 49 of *Operator Theory: Advances and Applications*. Birkhäuser Verlag, Basel, 1990.

[51] P. D. Lax, *Functional Analysis*. Pure and Applied Mathematics (New York), Wiley-Interscience [John Wiley & Sons], New York, 2002.

[52] M. Reed and B. Simon, *Methods of Modern Mathematical Physics I: Functional Analysis*. Academic Press, Inc., New York, second ed., 1980.

[53] Z. Wang, "Nonlinear model reduction based on the finite element method with interpolated coefficients: semilinear parabolic equations," *Numer. Methods Partial Differential Equations*, vol. 31, no. 6, pp. 1713–1741, 2015.

[54] K. Glover, R. F. Curtain, and J. R. Partington, "Realisation and Approximation of Linear Infinite-Dimensional Systems with Error Bounds," *SIAM Journal on Control and Optimization*, vol. 26, no. 4, pp. 863–898, 1988.

[55] P. Galán del Sastre and R. Bermejo, "Error Estimates of Proper Orthogonal Decomposition Eigenvectors and Galerkin Projection for a Aeneral Dynamical System Arising in Fluid Models," *Numerische Mathematik*, vol. 110, pp. 49–81, Jul 2008.

[56] I. C. Gohberg and M. G. Kreĭn, *Introduction to the theory of linear nonselfadjoint operators*. Translated from the Russian by A. Feinstein. Translations of Mathematical Monographs, Vol. 18, American Mathematical Society, Providence, R.I., 1969.

SECTION

2. CONCLUDING REMARKS AND FUTURE WORK

2.1. CONCLUDING REMARKS

We extended Brand's incremental SVD algorithm [27] to treat data expanded in basis functions from a Hilbert space. Many numerical methods for PDEs generate data of this form. Specifically, we reformulated Brand's matrix algorithm in a weighted norm setting to obtain the POD computations for a set of PDE simulation data. In this work, we extended Brand's algorithm to update the SVD and an error bound incrementally when a new column is added. We also considered time varying data by incorporating quadrature on the time integral into the incremental approach. Standard methods for computing the POD modes require storing the whole large dataset; in contrast, using an incremental SVD algorithm only requires storing one snapshot of the data at a time. Therefore, the incremental approach drastically reduces the memory requirement for computing the POD and error bound of the data. Furthermore, the computational cost of the incremental approach is also much lower than standard approaches. Moreover, by truncating small singular values (and corresponding singular vectors) during the incremental update, we reduce the computational cost of orthgonalizing the stored singular vectors. We also performed an error analysis of this algorithm by analyzing the effect of truncation at each step, and provided more insight into the accuracy of the algorithm with truncation and the choices of the two tolerances. We showed that the algorithm produces the exact SVD of a matrix $\tilde{U}$ such that $\|U - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^s, \mathbb{R}^m)} \leq e$, where U is the true data matrix, M is the weight matrix, and e is computed error bound. This error bound allows us to find error bounds for the incrementally computed singular values and singular vectors. We tested our approach

on finite element simulation data with the L^2 inner product for a 1D Burgers' equation, a 1D FitzHugh-Nagumo PDE system, and a 2D Navier-Stokes equation with various choices of the two truncation tolerances. For the small-scale computational cases, we compared the incremental SVD results with the exact SVD and found excellent agreement. For the 1D FitzHugh-Nagumo PDE, we found that the incrementally computed error bound was larger than the exact error and the error bound was small. Furthermore, the approximate singular values and dominant singular vectors were accurate. Also, our analysis and the numerical tests suggest that there is no benefit from choosing one algorithm tolerance different than the other. We also found that the incremental algorithm worked very well using a different inner product and also if we removed the average from the data.

2.2. FUTURE RESEARCH IDEAS

- In the first paper, we approximated the continuous time POD using a Riemann sum approximation for the integral. It would be interesting to develop and analyze incremental POD algorithms using more accurate approximations to the integral, such as the trapezoid rule.

- In the second paper, we analyzed the incremental SVD algorithm assuming all arithmetic is exact. Future work could include a rounding error analysis of the algorithm.

- As mentioned in the introduction to the second paper, there are other existing incremental SVD algorithms. It would be interesting to extend some of these algorithms to the weighted inner product case, perform an error analysis of the resulting algorithms, and thoroughly compare these new algorithms to the incremental algorithm developed in our work.

REFERENCES

[1] J. L. Lumley, "The structure of inhomogeneous turbulent flows," *Atmospheric Turbulence and Radio Wave Propagation*, 1967.

[2] K. Pearson, "LIII. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[3] H. Hotelling, "Analysis of a complex of statistical variables into principal components.," *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[4] K. Karhunen, "Zur spektraltheorie stochastischer prozesse," *Annales Academiae Scientarum Fennicae*, vol. 34, 1946.

[5] M. Loève, "Functions aléatoire de second ordre cr acad," *Sci., Paris*, vol. 220, 1945.

[6] D. Amsallem and C. Farhat, "Interpolation method for adapting reduced-order models and application to aeroelasticity," *AIAA Journal*, vol. 46, no. 7, pp. 1803–1813, 2008.

[7] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, Cambridge, second ed., 2012.

[8] C. W. Rowley, T. Colonius, and R. M. Murray, "Model reduction for compressible flows using POD and Galerkin projection," *Phys. D*, vol. 189, no. 1-2, pp. 115–129, 2004.

[9] H. T. Banks, R. C. H. del Rosario, and R. C. Smith, "Reduced-order model feedback control design: numerical implementation in a thin shell model," *IEEE Trans. Automat. Control*, vol. 45, no. 7, pp. 1312–1324, 2000.

[10] P. Benner, E. Sachs, and S. Volkwein, "Model order reduction for PDE constrained optimization," vol. 165, pp. 303–326, 2014.

[11] M. Gubisch and S. Volkwein, "POD for linear-quadratic optimal control," in *Model Reduction and Approximation: Theory and Algorithms* (P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, eds.), Philadelphia, PA: SIAM, 2017.

[12] M. Gunzburger, N. Jiang, and M. Schneier, "An ensemble-proper orthogonal decomposition method for the nonstationary Navier-Stokes equations," *SIAM J. Numer. Anal.*, vol. 55, no. 1, pp. 286–304, 2017.

[13] R. Ştefănescu, A. Sandu, and I. M. Navon, "POD/DEIM reduced-order strategies for efficient four dimensional variational data assimilation," *J. Comput. Phys.*, vol. 295, pp. 569–595, 2015.

[14] S. Qian, X. Lv, Y. Cao, and F. Shao, "Parameter estimation for a 2D tidal model with POD 4D VAR data assimilation," *Mathematical Problems in Engineering*, vol. 2016, 2016. Article ID 6751537.

[15] L. Sirovich, "Turbulence and the dynamics of coherent structures. I. Coherent structures," *Quarterly of Applied Mathematics*, vol. 45, no. 3, pp. 561–571, 1987.

[16] R. Pinnau, "Model reduction via proper orthogonal decomposition," in *Model order reduction: theory, research aspects and applications*, vol. 13 of *Math. Ind.*, pp. 95–109, Springer, Berlin, 2008.

[17] J. R. Singler, "Convergent snapshot algorithms for infinite-dimensional Lyapunov equations," *IMA J. Numer. Anal.*, vol. 31, no. 4, pp. 1468–1496, 2011.

[18] S. Volkwein, "Proper orthogonal decomposition: Theory and reduced-order modelling (lecture notes)," 2013.

[19] G. H. Golub and C. F. Van Loan, *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, third ed., 1996.

[20] N. J. Higham, *Accuracy and stability of numerical algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second ed., 2002.

[21] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, Cambridge, second ed., 2013.

[22] M. Fahl, "Computation of POD basis functions for fluid flows with Lanczos methods," *Math. Comput. Modelling*, vol. 34, no. 1-2, pp. 91–107, 2001.

[23] C. A. Beattie, J. Borggaard, S. Gugercin, and T. Iliescu, "A Domain Decomposition Approach to POD," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 6750–6756, Dec 2006.

[24] Z. Wang, B. McBee, and T. Iliescu, "Approximate partitioned method of snapshots for POD," *J. Comput. Appl. Math.*, vol. 307, pp. 374–384, 2016.

[25] C. Himpe, T. Leibner, and S. Rave, "Hierarchical approximate proper orthogonal decomposition," *arXiv preprint arXiv:1607.05210*, 2016.

[26] C. G. Baker, K. A. Gallivan, and P. Van Dooren, "Low-rank incremental methods for computing dominant singular subspaces," *Linear Algebra Appl.*, vol. 436, no. 8, pp. 2866–2888, 2012.

[27] M. Brand, *Incremental Singular Value Decomposition of Uncertain Data with Missing Values*, pp. 707–720. Computer Vision — ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part I, Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[28] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra Appl.*, vol. 415, no. 1, pp. 20–30, 2006.

[29] Y. Chahlaoui, K. Gallivan, and P. Van Dooren, "Recursive calculation of dominant singular subspaces," *SIAM J. Matrix Anal. Appl.*, vol. 25, no. 2, pp. 445–463, 2003.

[30] M. A. Iwen and B. W. Ong, "A distributed and incremental SVD algorithm for agglomerative data analysis on large networks," *SIAM J. Matrix Anal. Appl.*, vol. 37, no. 4, pp. 1699–1718, 2016.

[31] N. Mastronardi, M. Van Barel, and R. Vandebril, "A note on the recursive calculation of dominant singular subspaces," *Numer. Algorithms*, vol. 38, no. 4, pp. 237–242, 2005.

[32] N. Mastronardi, M. Van Barel, and R. Vandebril, "A fast algorithm for the recursive calculation of dominant singular subspaces," *J. Comput. Appl. Math.*, vol. 218, no. 2, pp. 238–246, 2008.

[33] Z. Drmac and A. K. Saibaba, "The discrete empirical interpolation method: Canonical structure and formulation in weighted inner product spaces," *arXiv preprint arXiv:1704.06606*, 2017.

[34] M. Tabandeh, M. Wei, and J. P. Collins, "On the symmetrization in POD-Galerkin model for linearized compressible flows," in *54th AIAA Aerospace Sciences Meeting*, p. 1106, 2016.

[35] G. Serre, P. Lafon, X. Gloerfelt, and C. Bailly, "Reliable reduced-order models for time-dependent linearized Euler equations," *J. Comput. Phys.*, vol. 231, no. 15, pp. 5176–5194, 2012.

[36] D. Amsallem, M. J. Zahr, and K. Washabaugh, "Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction," *Adv. Comput. Math.*, vol. 41, no. 5, pp. 1187–1230, 2015.

[37] B. Peherstorfer and K. Willcox, "Dynamic data-driven reduced-order models," *Comput. Methods Appl. Mech. Engrg.*, vol. 291, pp. 21–41, 2015.

[38] M. J. Zahr and C. Farhat, "Progressive construction of a parametric reduced-order model for PDE-constrained optimization," *Internat. J. Numer. Methods Engrg.*, vol. 102, no. 5, pp. 1111–1135, 2015.

[39] G. M. Oxberry, T. Kostova-Vassilevska, W. Arrighi, and K. Chand, "Limited-memory adaptive snapshot selection for proper orthogonal decomposition," *International Journal for Numerical Methods in Engineering*, vol. 109, no. 2, pp. 198–217, 2017.

[40] H. Fareed, J. Shen, J. R. Singler, and Y. Zhang, "Incremental proper orthogonal decomposition for PDE simulation data," *Computers & Mathematics with Applications*, vol. 75, no. 6, pp. 1942 – 1960, 2018.

[41] H. Fareed and J. R. Singler, "Error Analysis of an Incremental POD Algorithm for PDE simulation data," *arXiv preprint arXiv:1803.06313, submitted*.

[42] I. Gohberg, S. Goldberg, and M. A. Kaashoek, *Classes of Linear Operators. Vol. I*, vol. 49 of *Operator Theory: Advances and Applications*. Birkhäuser Verlag, Basel, 1990.

[43] P. D. Lax, *Functional Analysis*. Pure and Applied Mathematics (New York), Wiley-Interscience [John Wiley & Sons], New York, 2002.

[44] M. Reed and B. Simon, *Methods of Modern Mathematical Physics I: Functional Analysis*. Academic Press, Inc., New York, second ed., 1980.

[45] G. H. Golub and C. F. Van Loan, *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, fourth ed., 2013.

[46] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., *Templates for the solution of algebraic eigenvalue problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[47] W. Gander, M. J. Gander, and F. Kwok, *Scientific computing*, vol. 11 of *Texts in Computational Science and Engineering*. Springer, Cham, 2014.

[48] K. Kunisch and S. Volkwein, "Galerkin proper orthogonal decomposition methods for parabolic problems," *Numer. Math.*, vol. 90, no. 1, pp. 117–148, 2001.

[49] K. Kunisch and S. Volkwein, "Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics," *SIAM J. Numer. Anal.*, vol. 40, no. 2, pp. 492–515, 2002.

[50] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher, *Benchmark Computations of Laminar Flow Around a Cylinder*, pp. 547–566. Wiesbaden: Vieweg+Teubner Verlag, 1996.

[51] I. Akhtar, J. Borggaard, J. A. Burns, H. Imtiaz, and L. Zietsman, "Using functional gains for effective sensor location in flow control: a reduced-order modelling approach," *J. Fluid Mech.*, vol. 781, pp. 622–656, 2015.

[52] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele, "A hierarchy of low-dimensional models for the transient and post-transient cylinder wake," *J. Fluid Mech.*, vol. 497, pp. 335–363, 2003.

[53] J. R. Shewchuk, "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator," in *Applied Computational Geometry: Towards Geometric Engineering* (M. C. Lin and D. Manocha, eds.), vol. 1148 of *Lecture Notes in Computer Science*, pp. 203–222, Springer-Verlag, 1996.

[54] J. R. Shewchuk, "Triangle: A two-dimensional quality mesh generator and delaunay triangulator, version 1.6," 2005.

[55] T. Colonius and J. Freund, "POD analysis of sound generation by a turbulent jet," in *40th AIAA Aerospace Sciences Meeting & Exhibit*, p. 72, 2002.

[56] R. Zimmermann and S. G?rtz, "Non-linear reduced order models for steady aerodynamics," *Procedia Computer Science*, vol. 1, no. 1, pp. 165 – 174, 2010. ICCS 2010.

[57] R. Zimmermann, "A locally parametrized reduced-order model for the linear frequency domain approach to time-accurate computational fluid dynamics," *SIAM J. Sci. Comput.*, vol. 36, no. 3, pp. B508–B537, 2014.

[58] L. Peng and K. Mohseni, "Nonlinear model reduction via a locally weighted pod method," *International Journal for Numerical Methods in Engineering*, vol. 106, no. 5, pp. 372–396, 2016. nme.5124.

[59] E. A. Christensen, M. Brϕns, and J. N. Sϕrensen, "Evaluation of proper orthogonal decomposition-based decomposition techniques applied to parameter-dependent nonturbulent flows," *SIAM J. Sci. Comput.*, vol. 21, no. 4, pp. 1419–1434, 1999/00.

[60] I. Kalashnikova, M. F. Barone, S. Arunajatesan, and B. G. van Bloemen Waanders, "Construction of energy-stable projection-based reduced order models," *Appl. Math. Comput.*, vol. 249, pp. 569–596, 2014.

[61] D. Amsallem and J. Nordström, "Energy stable model reduction of neurons by nonnegative discrete empirical interpolation," *SIAM J. Sci. Comput.*, vol. 38, no. 2, pp. B297–B326, 2016.

[62] D. N. Daescu and I. M. Navon, "A Dual-Weighted Approach to Order Reduction in 4DVAR Data Assimilation," *Monthly Weather Review*, vol. 136, no. 3, pp. 1026–1041, 2008.

[63] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge Monographs on Mechanics, Cambridge University Press, Cambridge, second ed., 2012.

[64] M. F. Barone, I. Kalashnikova, D. J. Segalman, and H. K. Thornquist, "Stable Galerkin reduced order models for linearized compressible flow," *J. Comput. Phys.*, vol. 228, no. 6, pp. 1932–1946, 2009.

[65] V. M. Calo, Y. Efendiev, J. Galvis, and M. Ghommem, "Multiscale empirical interpolation for solving nonlinear PDEs," *J. Comput. Phys.*, vol. 278, pp. 204–220, 2014.

[66] C. Farhat, T. Chapman, and P. Avery, "Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models," *Internat. J. Numer. Methods Engrg.*, vol. 102, no. 5, pp. 1077–1110, 2015.

[67] X. Xie, D. Wells, Z. Wang, and T. Iliescu, "Numerical analysis of the Leray reduced order model," *J. Comput. Appl. Math.*, vol. 328, pp. 12–29, 2018.

[68] M. Mohebujjaman, L. G. Rebholz, X. Xie, and T. Iliescu, "Energy balance and mass conservation in reduced order models of fluid flows," *J. Comput. Phys.*, vol. 346, pp. 262–277, 2017.

[69] M. Gunzburger, N. Jiang, and M. Schneier, "An ensemble-proper orthogonal decomposition method for the nonstationary Navier-Stokes equations," *SIAM J. Numer. Anal.*, vol. 55, no. 1, pp. 286–304, 2017.

[70] T. Kostova-Vassilevska and G. M. Oxberry, "Model reduction of dynamical systems by proper orthogonal decomposition: error bounds and comparison of methods using snapshots from the solution and the time derivatives," *J. Comput. Appl. Math.*, vol. 330, pp. 553–573, 2018.

[71] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced basis methods for partial differential equations*. Springer, Cham, 2016.

[72] J. R. Singler, "New POD error expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs," *SIAM J. Numer. Anal.*, vol. 52, no. 2, pp. 852–876, 2014.

[73] A. Placzek, D.-M. Tran, and R. Ohayon, "A nonlinear POD-Galerkin reduced-order model for compressible flows taking into account rigid body motions," *Comput. Methods Appl. Mech. Engrg.*, vol. 200, no. 49-52, pp. 3497–3514, 2011.

[74] A. Corigliano, M. Dossi, and S. Mariani, "Model order reduction and domain decomposition strategies for the solution of the dynamic elastic-plastic structural problem," *Comput. Methods Appl. Mech. Engrg.*, vol. 290, pp. 127–155, 2015.

[75] B. Peherstorfer and K. Willcox, "Dynamic data-driven model reduction: adapting reduced models from incomplete data," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 3, p. 11, Mar 2016.

[76] O. T. Schmidt, "An efficient streaming algorithm for spectral proper orthogonal decomposition," *arXiv preprint arXiv:1711.04199*, 2017.

[77] R. Zimmermann, "A closed-form update for orthogonal matrix decompositions under arbitrary rank-one modifications," *arXiv preprint arXiv:1711.08235*, 2017.

[78] R. Zimmermann, B. Peherstorfer, and K. Willcox, "Geometric Subspace Updates with Applications to Online Adaptive Nonlinear Model Reduction," *SIAM J. Matrix Anal. Appl.*, vol. 39, no. 1, pp. 234–261, 2018.

[79] L. Giraud and J. Langou, "When modified Gram-Schmidt generates a well-conditioned set of vectors," *IMA J. Numer. Anal.*, vol. 22, no. 4, pp. 521–528, 2002.

[80] L. Giraud, J. Langou, M. Rozloznik, and J. van den Eshof, "Rounding error analysis of the classical Gram-Schmidt orthogonalization process," *Numer. Math.*, vol. 101, no. 1, pp. 87–100, 2005.

[81] L. Giraud, J. Langou, and M. Rozloznik, "The loss of orthogonality in the Gram-Schmidt orthogonalization process," *Comput. Math. Appl.*, vol. 50, no. 7, pp. 1069–1075, 2005.

[82] M. Rozloznik, M. Tuma, A. Smoktunowicz, and J. Kopal, "Numerical stability of orthogonalization methods with a non-standard inner product," *BIT*, vol. 52, no. 4, pp. 1035–1058, 2012.

[83] Z. Wang, "Nonlinear model reduction based on the finite element method with interpolated coefficients: semilinear parabolic equations," *Numer. Methods Partial Differential Equations*, vol. 31, no. 6, pp. 1713–1741, 2015.

[84] K. Glover, R. F. Curtain, and J. R. Partington, "Realisation and Approximation of Linear Infinite-Dimensional Systems with Error Bounds," *SIAM Journal on Control and Optimization*, vol. 26, no. 4, pp. 863–898, 1988.

[85] P. Galán del Sastre and R. Bermejo, "Error Estimates of Proper Orthogonal Decomposition Eigenvectors and Galerkin Projection for a Aeneral Dynamical System Arising in Fluid Models," *Numerische Mathematik*, vol. 110, pp. 49–81, Jul 2008.

[86] I. C. Gohberg and M. G. Kreĭn, *Introduction to the theory of linear nonselfadjoint operators*. Translated from the Russian by A. Feinstein. Translations of Mathematical Monographs, Vol. 18, American Mathematical Society, Providence, R.I., 1969.

VITA

Hiba Ghassan Fareed was born in Baghdad, Iraq, in 1979. She received her Bachelor's degree and Master's degree in Mathematics and computer application science from Al-Nahrain University, Baghdad, Iraq, in 2000 and 2002, respectively. Hiba worked as a programmer in Iraqi Ministry of Industry and Mineral – State Company for Constructional Industries from 2003 to 2005. In 2005, she joined Ministry of Higher Education and Scientific Research – Almustansiriya University – College of Science – Mathematics' Department as Assistant Lecturer. She was awarded a scholarship from the HCED Iraq for her PhD education in USA. She received her PhD in Mathematics from Missouri University of Science and Technology, Rolla, Missouri, USA, in May 2018.